Universidad Politécnica de Valencia

Departamento de

Informática de Sistemas y Computadores

Tesis de Master

Alta disponibilidad en servidores y optimización de recursos hardware a bajo coste.

Autor:
Aurelio Rubio Sapiña

Director:
Juan Vicente Capella Hernández

Valencia, Junio 2012

Resumen

Desde siempre los administradores de sistemas se han encontrado con "sentimientos contrapuestos" a la hora de seleccionar el sistema operativo sobre el que van a correr sus servicios, aunque uno quiera no todos los servicios pueden corrren sobre los mismos sistemas, por otro lado siempre ha sido una preocupación el hardware sobre el que van a correr, así como la duplicidad del mismo. A todo esto, si le añadimos el bajo porcentaje de uso que se le suele dar a bastantes servidores en el desempeño de sus funciones, la cantidad de espacio que necesitamos habilitar para ellos, la electronica necesaria para conectarlos todos y el consumo eléctrico que se produce, nos lleva a considerar alternativas como las propuestas en este documento para la optimización y reducción de costes, así como la mejora de disponibilidad de dichos servicios.

No solo en epocas de crisis como la actual una empresa necesita optimizar los costes en TIC, es más, todo aquel que ha trabajado como administrador de sistemas sabe que hacer entender a sus directivos la importancia de invertir en recursos informáticos es una tarea tan o más complicada que la instalción y configuración de clusters con virtualuzación y alta disponibilidad como el que vamos a presentar aquí. Es por eso que la propuesta que presentamos esta orientada a todas aquellas pequeñas/medianas empresas, que necesiten aportar flexibilidad y estabilidad a sus servicios a un bajo coste y que permita a sus administradores poder gestionar los recursos sacandoles más provecho y dependiendo menos de los fallos que puedan aparecer.

Índice General

Índice de Figuras

Índice de tablas

Capitulo 1

Introducción

Cada día es más habitual encontrarnos con que los servicios que debemos proporcionar deben cumplir con ciertos requisitos de bajo tiempo de respuesta, eficiencia, simplicidad, disponibilidad las 24 horas al día, los 7 días de la semana, todo el año. Es entonces cuando consideramos que ese determinado servicio (base de datos, página web, servicios de impresión, almacenamiento, VozIP) debe estar continuamente disponible, lo que implica que debamos aplicar determinadas técnicas tanto hardware como software para cumplir este objetivo.

Tal como se indica en [REDIRIS001] gracias a las tecnologías de virtualización y redes podemos construir arquitecturas geográficamente distribuidas, tolerantes a fallos con bajos tiempos de indisponibilidad.

Además, con la virtualización vamos a paliar, y en muchos casos eliminar, la infrautilización del hardware de los servidores, haciendo un uso más eficiente de los recursos del servidor.

La mejor manera que tenemos actualmente de unir todos estos servicios y poder ofrecer bajos tiempos de indisponibilidad así como sitemas tolerantes a fallos, es mediante el uso de clusters de ordenadores, donde un conjunto de dos o más máquinas que se caracterizan por mantener una serie de servicios compartidos y por estar constantemente monitorizándose entre sí [WIKI001].

En la actualidad la gran demanda de servicos informáticos y la necesidad de que estos esten disponibles el mayor tiempo posible hace que estas tecnologías sean cada vez más estudiadas y solicitadas en todo tipo de entornos, académicos, empresariales, en servicios públicos y privados.

Multitud de empresas ofrecen servicios de alta disponibilidad basados o no en virtualización, pero para aquellas empresas o servicios que requieran de estas tecnologías sin poder permitirse el excesivo coste que muchas veces conllevan este tipo de instalaciones, existen opciones de bajo costo basadas en software libre, como se propone en [HA001-2] y [VIRT001-3] por ejemplo.

En esta línea, la principal motivación que da origen a esta tesis de Máster es reducir la dependencia del hardware de nuestros sistemas servidores mediante la virtualización y a continuación aplicar técnicas de clusters de alta disponibilidad para reducir los tiempos de indisponibilidad de los mismos. Siguiendo estos pasos, vamos a unir todas estas tecnologías en un sistema que nos permita disponer de servicios que estén activos el mayor tiempo posible y con un buen aprovechamiento de los recursos que disponemos, todo ello a un bajo coste. Para ello vamos a realizar un estudio comparativo de las herramientas de virtualización que nos permitirán hacer todo nuestro sistema menos dependiente de los fallos en el hardware del cual disponemos y mejor optimización del mismo. A continuación realizaremos un estudio de las diferentes alternativas que podemos encontrarnos en el momento que intentemos otorgar a nuestros sistemas esa etiqueta de Alta Disponibilidad [HA], centrándonos principalmente en reducir los tiempos de indisponibilidad en los servidores.

Virtualización

Cuando hablamos de **virtualización** estamos haciendo referencia a la abstracción de los recursos de una computadora, mediante la creación de una capa entre el hardware de la máquina física (host) y el sistema operativo de la máquina virtual (virtual machine, guest), siendo un medio para crear una versión virtual de un dispositivo o recurso, como un servidor, un dispositivo de almacenamiento, una red o incluso un sistema operativo, donde se divide el recurso en uno o más entornos de ejecución.

Un ejemplo de virtualización muy conocido son las maquinas virtuales, que generalmente son un sistema operativo completo que corre como si estuviera instalado en una plataforma de hardware autónoma, donde para que el sistema operativo "guest" funcione, la simulación debe ser lo suficientemente robusta (dependiendo del tipo de virtualización).

Esta capa de software maneja, gestiona y arbitra los cuatro recursos principales de una computadora (CPU, Memoria, Red, Almacenamiento) y así podrá repartir dinámicamente dichos recursos entre todas las máquinas virtuales definidas en el host.

El auge de la virtualización ha sido impulsado sobre todo por las siguientes características:

Mejora en la utilización de los recursos.

Eficiencia energética.

Reducción significativa del espacio físico necesario.

Recuperación de desastres y mejora de la disponibilidad

Reducción general de costes de operación.

Actualmente se distinguen tres tipos de virtualización:

Virtualización de Hardware.

Actualmente la más usada y la que da lugar al estudio que realizamos en el cual vamos a virtualizar servidores.

Virtualización completa (*Full virtualization*): En esta virtualización es donde la máquina virtual simula un hardware suficiente para permitir un sistema operativo "huésped" sin modificar (uno diseñado para la misma CPU) para correr de forma aislada. Típicamente, muchas instancias pueden correr al mismo tiempo.

Podemos citar como ejemplos de estos sistemas de virtualización entre otros:

- VMware Workstation y VMware Server. [VIRT004]
- Windows Server 2008 R2 Hyper-V. [VIRT005]
- Microsoft Enterprise Desktop Virtualization (MED-V). [VIRT006]
- VirtualBox. [VIRT007]
- Parallels Desktop. [VIRT008]
- OpenVZ. [VIRT009]
- XenServer . [VIRT001]
- Microsoft Virtual PC. [VIRT010]
- Virtual Iron, Adeos, Mac-on-Linux, Win4BSD, Win4Lin Pro, y z/VM, Oracle VM

Virtualización parcial (*Partial virtualization*): "Address Space Virtualization". La máquina virtual simula múltiples instancias de gran parte (pero no de todo) del entorno subyacente del hardware, particularmente los espacios de direcciones. Tal entorno acepta compartir recursos y alojar procesos, pero no permite instancias separadas de sistemas operativos "huésped". Aunque no es vista como dentro de la categoría de máquina virtual, históricamente éste fue un importante acercamiento lo usaron en sistemas como CTSS, el experimental IBM M44/44X [WIKI002], y podría mencionarse que en sistemas como OS/VS1, OS/VS2 y MVS [WIKI003].

Virtualización asistida por Hardware (*ParaVirtualization*): Virtualización asistida por Hardware son extensiones introducidas en la arquitectura de procesador x86 para facilitar las tareas de virtualización al software corriendo sobre el sistema. Son cuatro los niveles de privilegio o "anillos" de ejecución en esta arquitectura, desde el cero o de mayor privilegio, que se destina a las

operaciones del kernel de SO, al tres, con privilegios menores que es el utilizado por los procesos de usuario, en esta nueva arquitectura se introduce un anillo interior o ring -1 que será el que un *hypervisor* o *Virtual Machine Monitor* usará para aislar todas las capas superiores de software de las operaciones de virtualización.

La Paravirtualización es una técnica moderna ejecución virtual que consiste en permitir algunas llamadas directas al hardware mermando así la penalización en rendimiento que la ejecución 100% virtual implica. Esto es posible gracias a características que los procesadores modernos tienen, p.e: Intel tiene Intel VT [VIRT011] y AMD tiene AMD-V [VIRT012]. Estas APIs ofrecen instrucciones especiales que el software de virtualización puede emplear para permitir una ejecución más eficiente.

Virtualización de Sistemas (LPAR): Nos permite particionar un sistema físico en multiples sistemas lógicos o "virtuales" para ejecutar diferentes sistemas en cada una de las particiones, usando para cada una de estas particiones hardware dedicado o compartido según necesidades. En algunos de los casos se va a permitir reconfigurar las particiones dinámicamente pudiendo añadir y/o eleminar recursos de cada una de las particiones [VIRT016].

Virtualización de Aplicaciones

La virtualización de aplicaciones permite la gestión, el mantenimiento y almacenamiento centralizado para las aplicaciones, además de que se distribuyen sobre la red y se ejecutan localmente en las máquinas cliente. Con ello entre otras cosas se consigue un entorno informático más flexible, que permite una mayor y más rápida respuesta de las organizaciones ante un cambio en sus necesidades o condiciones de mercado.

Además, con la virtualización de aplicaciones vamos a tener la ventaja de poder ejecutar las aplicaciones virtualizadas independientemente del entorno y sistemas donde la ejecutemos. Como ejemplo cabe citar XenApp [VIRT013].

Virtualización de Escritorios

Como respuesta a los contínuos cambios los departamentos de TI han de atender las exigencias de los usuarios en cuanto a más flexibilidad, así como su deseo y su necesidad de acceder a las aplicaciones corporativas y a sus datos desde cualquier lugar, con cualquier dispositivo, en cualquier momento y de manera segura.

Con la virtualización de puestos de trabajo, los departamentos de TI pueden replantearse partiendo de cero su manera de provisionar equipos y administrar estrategias de BYO (*Bring*

Your Own,) efectivas [VIRT015].

La **virtualización de escritorios** es un término relativamente nuevo, introducido en la década de los 90, que describe el proceso de separación entre el escritorio, que engloba los datos y programas que utilizan los usuarios para trabajar, de la máquina física. El escritorio "virtualizado" es almacenado remotamente en un servidor central en lugar de en el disco duro del ordenador personal. Esto significa que cuando los usuarios trabajan en su escritorio desde su portátil u ordenador personal, todos sus programas, aplicaciones, procesos y datos se almacenan y ejecutan centralmente, permitiendo a los usuarios acceder remotamente a sus escritorios desde cualquier dispositivo capaz de conectarse remotamente al escritorio, tales como un portátil, PC, smartphone o cliente ligero.

La experiencia que tendrá el usuario está orientada para que sea idéntica a la de un PC estándar.

La virtualización del escritorio proporciona muchas de las ventajas de un servidor de terminales, además de poder proporcionar a los usuarios mucha más flexibilidad, como por ejemplo, cada uno

puede tener permitido instalar y configurar sus propias aplicaciones sin interferir con el resto de usuarios.

Las principales ventajas que ofrece este tipo de virtualización son:

- Aumenta la seguridad de los escritorios y disminuye los costes de soporte.
- Reduce los costes de ampliación/renovación de PC's, (parte cliente) que pasa de 1-3 años a 5-6años. Debe considerarse que parte de los costes de hardware se trasladan a la parte de servidores.
- Acceso a los escritorios y su contenido desde cualquier ubicación.

Aunque siempre hay que tener en cuenta que VDI se ajusta para cada cliente, pero no para todos los escritorios [VIRT014].

Clusters

Un cluster 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. Los cluster permiten aumentar la escabilidad, disponibilidad y fiabilidad.

Un cluster puede presentarse como una solución de especial interés sobre todo a nivel de empresas, las cuales pueden aprovecharse de estas especiales características de computación para mantener sus equipos actualizados por un precio bastante más económico que el que les supondría actualizar todos sus equipos informáticos y con unas capacidades de computación y disponibilidad.

Componentes de un Cluster

Un sistema cluster esta formado por diversos componentes hardware y software:

Nodos: Cada una de las máquinas que componen el cluster, pueden ser desde simples ordenadores personales a servidores dedicados, conectados por una red. Por regla general los nodos deben tener características similares: arquitectura, componentes, sistema operativo.

Sistemas Operativos: Se utilizan sistemas operativos de tipo servidor con características de multiproceso y multiusuario, así como capacidad para abstracción de dispositivos y trabajo con interfaces IP virtuales.

Middleware de Cluster: Es el software que actúa entre el sistema operativo y los servicios o aplicaciones finales. Es la parte fundamental del cluster donde se encuentra la lógica del mismo.

Conexiones de red: Los nodos del cluster pueden conectarse mediante una simple red Fast Ethernet o utilizar tecnologías de red avanzadas como Gigabit Ethernet, Infiniband, Myrinet, SCI, etc.

Protocolos de comunicación: Definen la intercomunicación entre los nodos del cluster.

Sistema de almacenamiento: El almacenamiento puede ir desde sistemas comunes de almacenamiento interno del servidor a redes de almacenamiento compartido NAS o SAN.

Servicios y aplicaciones: Son aquellos servicios y aplicaciones a ejecutar sobre el cluster habitualmente.

Tipos de Clusters

Clusters de Alta disponibilidad (HA, high availability)

Los clusters de alta disponibilidad tienen como propósito principal brindar la máxima

disponibilidad de los servicios que ofrecen. Esto se consigue mediante software que monitoriza constantemente el cluster, detecta fallos y permite recuperarse frente a los mismos.

Clusters de Alto Rendimiento (HP, high performance)

Estos clusters se utilizan para ejecutar programas paralelizables que requieren de gran capacidad computacional de forma intensiva. Son de especial interés para la comunidad científica o industrias que tengan que resolver complejos problemas o simulaciones.

Utilizando clustering, podemos crear hoy en día supercomputadores con una fracción del coste de un sistema de altas prestaciones tradicional.

Clusters de Balanceo de Carga (LB, Load Balancing)

Este tipo de cluster permite distribuir las peticiones de servicio entrantes hacia un conjunto de equipos que las procesa. Se utiliza principalmente para servicios de red sin estado, como un servidor web o un servidor de correo electrónico, con altas cargas de trabajo y de tráfico de red.

Las características más destacadas de este tipo de cluster son su robustez y su alto grado de escalabilidad.

Alta Disponibilidad

Cuando hablamos de **Alta disponibilidad** (*High availability*) hacemos referencia a un protocolo de diseño del sistema y su implementación asociada que asegura un determinado grado de continuidad operacional durante un período de medición dado. Disponibilidad se refiere a la habilidad de la comunidad de usuarios para acceder al sistema, utilizar sus servicios, lanzar nuevos trabajos, actualizar o alterar trabajos existentes o recoger los resultados de trabajos previos. Si un usuario no puede acceder al sistema se dice que está no disponible. El término tiempo de inactividad (*downtime*) es usado para definir cuándo el sistema no está disponible.

Podemos diferenciar entre tiempo de inactividad planificado (aquel que es imprescindible por actualizaciones del sistema, configuraciones y reinicios) y el tiempo de inactivadad no planificado que surgen a causa de algún evento físico, tales como fallos en el hardware o anomalías ambientales. Ejemplos de eventos con tiempos de inactividad no planificados incluyen fallos de potencia, fallos en los componentes de CPU o RAM, una caída por recalentamiento, una ruptura lógica o física en las conexiones de red, rupturas de seguridad catastróficas o fallos en el sistema operativo, aplicaciones y middleware. Muchos puestos computacionales excluyen tiempo de inactividad planificado de los cálculos de disponibilidad, asumiendo, correcta o incorrectamente, que el tiempo de actividad no planificado tiene poco o ningún impacto sobre los usuarios. Excluyendo tiempo de inactividad planificado, muchos sistemas pueden reclamar tener alta disponibilidad , lo cual da la ilusión de disponibilidad continua. Sistemas que exhiben verdadera disponibilidad continua son comparativamente raros y caros, y ellos tienen diseños cuidadosamente implementados que eliminan cualquier punto de fallo y permiten que el hardware, la red, el sistema operativo, middleware y actualización de aplicaciones, parches y reemplazos se hagan en línea.

Por otro lado, la **Disponibilidad** es usualmente expresada como un porcentaje del tiempo de funcionamiento en un año dado.

En un año dado, el número de minutos de tiempo de inactividad no planeado es registrado para un sistema, el tiempo de inactividad no planificado agregado es dividido por el número total de minutos en un año (aproximadamente 525.600) produciendo un porcentaje de tiempo de inactividad; el complemento es el porcentaje de tiempo de funcionamiento el cual es lo que denominamos como disponibilidad del sistema. Valores comunes de disponibilidad, típicamente enunciado como número de "nueves" para sistemas altamente disponibles son:

disponibilidad = t.disponible / (t.disponible + t.inactivo)

99,9% = 43.8 minutos/mes u 8,76 horas/año ("tres nueves")
99,99% = 4.38 minutos/mes o 52.6 minutos/año ("cuatro nueves")
99,999% = 0.44 minutos/mes o 5.26 minutos/año ("cinco nueves")

Porcentaje de disponibilidad	Tiempo de Interrupción Anual	Tiempo de Interrupción Semanal
98 %	7,3 días	3,3 horas
99 %	3,6 días	1,7 horas
99,9 %	8,8 horas	10 minutos
99,99 %	52,5 minutos	1 minuto
99,999 %	5,3 minutos	6 segundos
99,9999 %	31,5 segundos	0,6 segundos

Es de hacer notar que tiempo de funcionamiento y disponibilidad no son sinónimos. Un sistema puede estar en funcionamiento y no disponible como en el caso de un fallo de red. Se puede apreciar que estos valores de disponibilidad son visibles mayormente en documentos de ventas o marketing, en lugar de ser una especificación técnica completamente medible y cuantificable.

También tenemos otros conceptos tales como el **Tiempo de recuperación** que esta cercanamente relacionado con la disponibilidad y es el tiempo total requerido para un apagón planificado o el tiempo requerido para la recuperación completa de un apagón no planificado. El Tiempo de recuperación puede ser infinito con ciertos diseños y fallos del sistema, recuperación total es imposible. Uno de tales ejemplos es un incendio o inundación que destruye un centro de datos y sus sistemas cuando no hay un centro de datos secundario para recuperación frente a desastres.

Otro concepto relacionado es **disponibilidad de datos**, que es el grado para el cual las bases de datos y otros sistemas de almacenamiento de la información que registran y reportan fielmente transacciones del sistema. Especialistas de gestión de la información frecuentemente enfocan separadamente la disponibilidad de datos para determinar perdida de datos aceptable o actual con varios eventos de fracasos. Algunos usuarios pueden tolerar interrupciones en el servicio de aplicación pero no perdida de datos

Paradójicamente, añadiendo más componentes al sistema total puede socavar esfuerzos para lograr alta disponibilidad. Esto es debido a que sistemas complejos tienen inherentemente más puntos de fallos potenciales y son más difíciles de implementar correctamente. La mayoría de los sistemas altamente disponibles extraen a un patrón de diseño simple: un sistema físico multipropósito simple de alta calidad con redundancia interna comprensible ejecutando todas las funciones interdependientes emparejadas con un segundo sistema en una localización fisíca separada.

Objetivos y aportaciones de la tesis de Master

El objetivo principal de este proyecto es el estudio, diseño, implementación real y experimentación de un sistema de Alta disponibilidad para servidores virtualizados, con el que pretendemos dotar a los sistemas que implantemos de independencia del hardware sobre el que se ejecuta y a su vez de unos tiempos de recuperación e indisponibilidad reducidos. Todo ello con una reducción de costes importante en comparación con los sistemas comerciales existentes, reduciendo dichos costes tanto en su componente hardware como software, mejorando la flexibilidad del sistema.

Además se pretende realizar un estudio exhaustivo de las tecnologías para la implementación de los diferentes componentes del mismo, comparándolas muchas de ellas de forma práctica,

identificando la mejor combinación en cada caso de cara al sistema completo propuesto y considerando diferentes ámbitos de aplicación.

Desarrollo de la tesis de Master

La presente tesina se estructura en 5 capítulos en los que se presenta el trabajo realizado y se desarrollan las aportaciones antes citadas.

En el capítulo 2 se refleja el resultado del estudio a fondo realizado sobre las diferentes tecnologías empleadas para la implementación del sistema deseado (Almacenamiento, Virtualización, Alta disponibilidad).

En el capítulo 3 se expone el sistema propuesto, que presenta las ventajas que se han ido citando, constituye una de las principales aportaciones de la tesina.

En el capítulo 4 se presenta la metodología seguida en la fase de experimentación, se presenta la configuración/implementación del sistema planteado en la presente tesina, presentándose su estructura y funcionamiento. Asimismo se presentan algunas de las implementaciones realizadas para la realización de las experimentaciones, describiendo los escenarios y los resultados obtenidos en diversos campos.

Finalmente, el capítulo 5 presenta las conclusiones, las aportaciones y se exponen las líneas de trabajo futuras.

Capitulo 2

Estudio comparativo de los componentes del sistema

En este apartado se van a estudiar los distintos componentes de un cluster de alta disponibilidad de máquinas virtuales, presentando y comparando las tecnologías más extendidas hasta el momento. Estos componentes son por un lado el sistema de almacenamiento, parte muy importante en estos sistemas debido a que representa el principal cuello de botella en la eficiencia del mismo, y donde al igual que en su rendimiento, hay que tener muy presente su coste que suele ser uno de los elevados de todo lo propuesto. Por otro lado, se presentan las tecnologías más recientes en cuanto a virtualización, que nos permitirán obtener el mejor resultado en cuanto a rendimiento y majeno de los sistemas a virtualizar y la mayor independencia del hardware posible. En cuanto a tecnologías de clusters que aporten a nuestro sistema alta disponibilidad, vamos a estudiar aquellas que mejor gestionen los recursos propuestos al menor coste.

Almacenamiento

Empezaremos el estudio con los sitemas de almacenamiento, dentro de los cuales estudiaremos tanto los sitemas de almacenamiento propiamente dichos como los discos duros y sus principales sistemas de ficheros, evaluando en cada caso las mejores opciones para nuestros propósitos, teniendo siempre presente el coste de cada uno de las opciones propuestas y su rendimiento.

Hardware. Sistemas de Almacenamiento

SAN

Una **SAN** (*Storage Area Network*) es una red de almacenamiento dedicada que proporciona acceso de nivel de bloque a LUNs. Un LUN, o número de unidad lógica, es un disco virtual proporcionado por la SAN. El administrador del sistema tiene el mismo acceso y los derechos a la LUN como si fuera un disco directamente conectado a la misma. El administrador puede particionar y formatear el disco en cualquier medio que él elija.
Una SAN principalmente, está basada en tecnología **fibre channel** y más recientemente en **iSCSI**.
Una SAN se puede considerar una extensión de Direct Attached Storage (DAS).
Tanto en SAN como en DAS, las aplicaciones y programas de usuarios hacen sus peticiones de datos al sistema de ficheros directamente.

NAS

NAS (*Network Attached Storage*) es el nombre dado a una tecnología de almacenamiento dedicada a compartir la capacidad de almacenamiento de un computador (Servidor) con ordenadores personales o servidores clientes a través de una red (normalmente TCP/IP), haciendo uso de un Sistema Operativo optimizado para dar acceso con los protocolos CIFS, NFS o FTP.

Los protocolos de comunicaciones NAS son basados en ficheros por lo que el cliente solicita el fichero completo al servidor y lo maneja localmente, están por ello orientados a información almacenada en ficheros de pequeño tamaño y gran cantidad.

DAS

Direct Attached Storage (*DAS*) es el método tradicional de almacenamiento y el más sencillo. Consiste en conectar el dispositivo de almacenamiento directamente al servidor o estación de

trabajo, es decir, físicamente conectado al dispositivo que hace uso de él.

Tanto en DAS como en SAN (Storage Area Network), las aplicaciones y programas de usuarios hacen sus peticiones de datos al sistema de ficheros directamente. La diferencia entre ambas tecnologías reside en la manera en la que dicho sistema de ficheros obtiene los datos requeridos del almacenamiento. En una DAS, el almacenamiento es local al sistema de ficheros, mientras que en una SAN, el almacenamiento es remoto. En el lado opuesto se encuentra la tecnología NAS (Network-attached storage), donde las aplicaciones hacen las peticiones de datos a los sistemas de ficheros de manera remota

DRBD

DRBD (*Distributed Replicated Block Device*) es una paquete de software que nos permite crear una especie de RAID1 entro dos discos/sistemas conectados a través de la red/lan, sincronizando los datos entre las diferentes particiones de los dos servidores diferentes, tal como se muestra en la figura 1 [DRBD001].

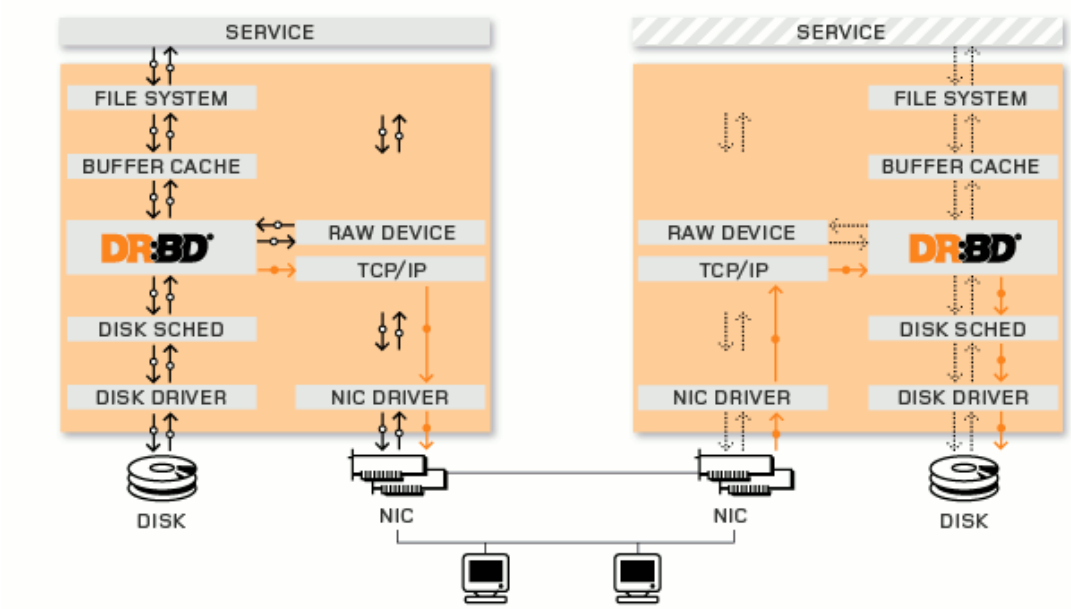

Figura 1: Estructura de un sistema con DRBD

Comparativa

El opuesto a NAS es la conexión DAS (Direct Attached Storage) mediante conexiones SCSI o la conexión SAN (Storage Area Network) por fibra óptica, en ambos casos con tarjetas de conexión especificas de conexión al almacenamiento. Estas conexiones directas (DAS) son por lo habitual dedicadas.

En la tecnología NAS, las aplicaciones y programas de usuario hacen las peticiones de datos a los sistemas de ficheros de manera remota mediante protocolos CIFS y NFS, el almacenamiento es local al sistema de ficheros. Sin embargo, DAS y SAN realizan las peticiones de datos directamente al sistema de ficheros, como se muestra en la figura 3.

Las ventajas del NAS sobre la conexión directa (DAS) son la capacidad de compartir las unidades, un menor coste, la utilización de la misma infraestructura de red y una gestión más sencilla. Por el contrario, NAS tiene un menor rendimiento y fiabilidad por el uso compartido de las comunicaciones.

Una SAN se puede considerar una extensión de Direct Attached Storage (DAS). Donde en DAS hay un enlace punto a punto entre el servidor y su almacenamiento, una SAN permite a varios servidores acceder a varios dispositivos de almacenamiento en una red compartida. Tanto en SAN como en DAS, las aplicaciones y programas de usuarios hacen sus peticiones de datos al sistema de ficheros directamente. La diferencia reside en la manera en la que dicho sistema de ficheros obtiene los datos requeridos del almacenamiento. En DAS, el almacenamiento es local al sistema de ficheros, mientras que en SAN, el almacenamiento es remoto. SAN utiliza diferentes protocolos de acceso como Fibre Channel y Gigabit Ethernet. En el lado opuesto se encuentra la tecnología Network-attached_storage (NAS), donde las aplicaciones hacen las peticiones de datos a los sistemas de ficheros de manera remota mediante protocolos CIFS y Network File System (NFS), ver figura 2.

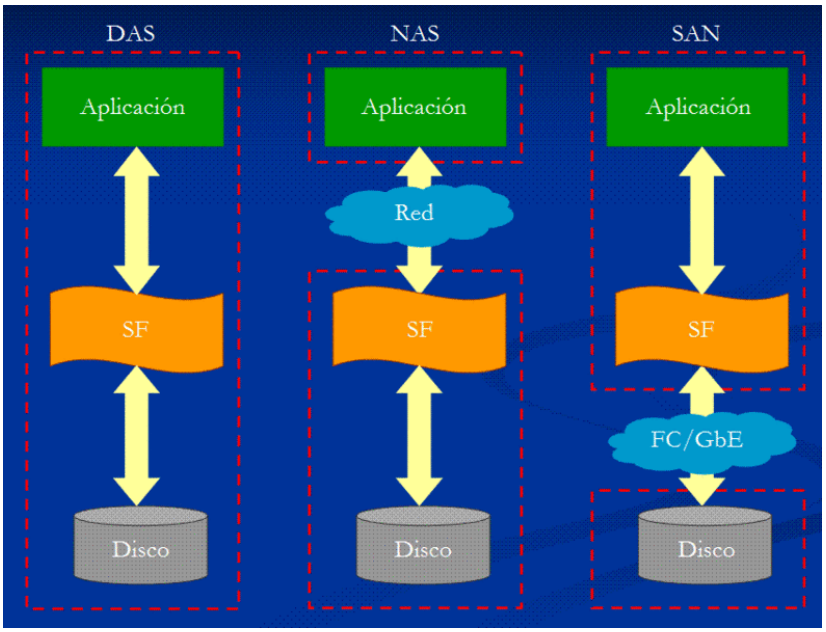

Figura 2: Estructura de los Sistemas de Almacenamiento

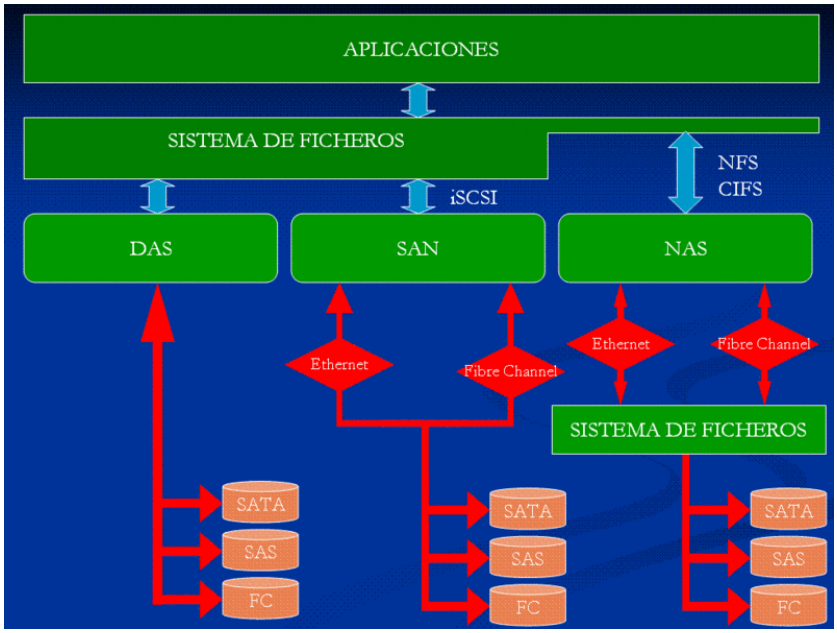

Figura 3: Esquema de acceso a los Sistemas de Almacenamiento

No obstante, pese a que DRBD no es exactamente hardware de almacenamiento en red, si que presenta varias funcionalidades que lo situan como un buen candidato para sustituir cualquiera de los métodos anteriores, ya que conseguirá reducir costes respecto a los sistemas SAN y aumentar rendimiento respecto a los sistemas NAS

Hardware. Discos Duros

En cuanto a las diferentes tecnologías de discos duros, estudiaremos las principales diferencias entre los sistemas báscios SAT, los sistemas de alto rendimiento SAS y los nuevos sistemas SSD basados en memoria no volátiles tipo flash y con unos tiempos de acceso muy superiores a cualquier otra tecnología usada hasta estos momentos, viendo en cada uno de los casos sus tiempos de acceso, los mejores campos de aplicación y porsupuesto su coste.

SAS

Serial Attached SCSI [HD001], es una interfaz de transferencia de datos en serie, sucesor del SCSI (Small Computer System Interface) paralelo, aunque sigue utilizando comandos SCSI para interaccionar con los dispositivos SAS. Aumenta la velocidad, permite la conexión y desconexión de forma rápida.

La primera versión apareció a finales de 2003, SAS 300, que conseguía un ancho de banda de 3Gb/s, lo que aumentaba ligeramente la velocidad de su predecesor , el SCSI Ultra 320MB/s (2,560 Gb/s). La siguiente evolución, SAS 600, consigue una velocidad de hasta 6Gb/s, ver especificaciones en la tabla 1.

Una de las principales características es que aumenta la velocidad de transferencia al aumentar el número de dispositivos conectados, es decir, puede gestionar una tasa de transferencia constante para cada dispositivo conectado, además de terminar con la limitación de 16 dispositivos existente en SCSI.

Además, el conector es el mismo que en la interfaz SATA y permite utilizar estos discos duros, para aplicaciones con menos necesidad de velocidad, ahorrando costos. Por lo tanto, los discos

SATA pueden ser utilizados por controladoras SAS pero no a la inversa, una controladora SATA no reconoce discos SAS.

Especificaciones Técnicas	Serial Attached SCSI
Prestaciones	Full-Duplex con Link Aggregation (ancho de banda 24 Gb/sec)
	3.0 Gb/sec (introducido) (6.0 Gb/s planeado)
Conectividad	8 metros de cable externo
	128 dispositivos Expansores de puerto (16K+ dispositivos totales)
	SAS a SATA compatibilitdad
Disponibilidad	Dual-port HDDs
	Multi-Initiator punto a punto
Driver	Software transparente con SCSI

Tabla 1: Especificaciones técnicas de SAS

SATA

Serial ATA [HD002], es una interfaz de transferencia de datos entre la placa base y algunos dispositivos de almacenamiento, como puede ser el disco duro, lectores y regrabadores de CD/DVD/BR, etc. Serial ATA sustituye a la tradicional **Parallel ATA** o P-ATA. SATA proporciona mayores velocidades, mejor aprovechamiento cuando hay varias unidades, mayor longitud del cable de transmisión de datos y capacidad para conectar unidades al instante.

La primera generación tiene una transferencias de 150 MB por segundo, también conocida por **SATA 150 MB/s** o Serial ATA-150. Actualmente se comercializan dispositivos **SATA II**, a 300 MB/s, también conocida como Serial ATA-300 y los **SATA III** con tasas de transferencias de hasta 600 MB/s.

Las Unidades que soportan la velocidad de 3Gb/s son compatibles con un bus de 1,5 Gb/s.

En la siguiente tabla se muestra el cálculo de la velocidad real de SATAI 1.5 Gb/s, SATAII 3 Gb/s y SATAIII 6 Gb/s:

SSD

Una **unidad de estado sólido** [HD003] es un dispositivo de almacenamiento de datos que usa una memoria no volátil, como la memoria flash. En comparación con los discos duros tradicionales, las unidades de estado sólido son menos susceptibles a golpes, son prácticamente inaudibles y tienen un menor tiempo de acceso y de latencia. Los SSD hacen uso de la misma interfaz que los discos duros y por tanto son fácilmente intercambiables sin tener que recurrir a adaptadores o tarjetas de expansión para compatibilizarlos con el equipo.

Los dispositivos de estado sólido que usan flash tienen varias ventajas únicas frente a los discos duros mecánicos:

- Arranque más rápido.
- Gran velocidad de lectura/escritura.
- Baja latencia de lectura y escritura, cientos de veces más rápido que los discos mecánicos.
- Menor consumo de energía y producción de calor. Resultado de no tener elementos mecánicos.
- Sin ruido. La misma carencia de partes mecánicas los hace completamente inaudibles.
- Rendimiento determinístico. A diferencia de los discos duros mecánicos, el rendimiento de los SSD es constante y determinístico a través del almacenamiento entero. El tiempo de "búsqueda" constante.

Los dispositivos de estado sólido que usan flash tienen también varias desventajas:

- Precio. Los precios de las memorias flash son considerablemente más altos en relación precio/gigabyte.
- Menor recuperación. Después de un fallo físico se pierden completamente los datos pues la celda es destruida, mientras que en un disco duro normal que sufre daño mecánico los datos son frecuentemente recuperables usando ayuda de expertos.
- Vida útil. En cualquier caso, reducir el tamaño del transistor implica reducir la vida útil de las memorias NAND, se espera que esto se solucione con sistemas utilizando memristores (memory resistor).

Comparativa

Si medimos **MB/seg** comprobaremos que los discos SAS son “solo”, un 25% o 30% más rápidos que los SATA.

Hablando de **IOPS** (operaciones de entrada/salida por segundo), un disco SATA, difícilmente llega al 20% de lo que ofrece un SAS de forma sostenida.

Para ciertos usos lo importante puede ser tener discos con mucha transferencia MB/seg:

- Streaming, modelado 3D, Backup…

Para los usos anteriores, un disco SATA es un magnífica elección, dado que su coste "por MB" es menor y una pérdida de rendimiento del 25% o 30% con respecto al SAS puede ser justificable en función de la aplicación.

- Bases de datos, Virtualización, Mail Server DB (Exchange)…

Con aplicaciones y sistemas operativos, donde vamos a pedir mil y un datos de forma concurrente al disco, sin duda el disco a seleccionar es SAS. Un rendimiento hasta 5 veces mayor justifica los costes.

Además los discos SAS, disponen de otras características como Full duplex interface (half duplex en SATA), tiempos medios de busqueda menores y más longevidad

Por otro lado, los discos SAS suelen ofrecer hasta 1,2 millones de horas de tiempo medio entre fallos trabajando las 24 horas del día, mientras los SATA ofrecen alrededor de 1 millón de horas pero asignandoles una carga de alrededor de 8 horas diárias de trabajo.

Los SSD, hoy por hoy, tienen como principales inconvenientes la durabilidad y el coste, pero en el momento en que se solucionen, es casi inevitable que empiecen a copar el mercado de disco de alto rendimiento.

Software. Sistemas de ficheros

Un sistema de archivos define todo lo relativo a la organización y gestión de archivos de computadora, además de los datos que estos contienen, para hacer más fácil la tarea localización y uso. Los sistemas de archivos más comunes utilizan dispositivos de almacenamiento de datos (Disco Duros, CDS, Floppys, USB Flash , etc..) e involucran el mantenimiento de la localización física de los archivos.

Más formalmente, un sistema de archivos es un conjunto de tipo de datos abstractos que son implementados para el almacenamiento, la organización jerárquica, la manipulación, el acceso, el direccionamiento y la recuperación de datos. Los sistemas de archivos comparten mucho en común con la tecnología de las bases de datos.

En general, los sistemas operativos tienen su propio sistema de archivos. En ellos, los sistemas de archivos pueden ser representados de forma textual (ej.: el shell de DOS) o gráficamente (p.ej.:

Explorador de archivos en Windows) utilizando un gestor de archivos.

El software del sistema de archivos es responsable de la organización de estos sectores en archivos y directorios, además mantiene un registro de qué sectores pertenece a qué archivos y cuáles no han sido utilizados.

Los sistemas de archivos pueden ser clasificados comunmente en tres categorías: sistemas de archivo de disco, sistemas de archivos de red y sistemas de archivos de propósito especial.

Sistemas de archivo de disco. Tipo especial de sistema de archivos diseñado para el almacenamiento, acceso y manipulación de archivos en un dispositivo de almacenamiento.

Son sistemas de archivos de disco: EFSa, EXT2, EXT3, FAT (sistema de archivos de DOS y algunas versiones de Windows), UMSDOS, FFS, Fossil ,HFS (para Mac OS), HPFS, ISO 9660 (sistema de archivos de sólo lectura para CD-ROM), JFS, kfs, MFS (para Mac OS), Minix, NTFS (sistema de archivos de Windows NT, XP y Vista), OFS, ReiserFS, Reiser4, UDF (usado en DVD y

en algunos CD-ROM), UFS, XFS, etc.

Sistemas de archivos de red. Tipo especial de sistema de archivos diseñado para acceder a sus archivos a través de una red. Este sistema de puede clasificar en dos: los sistemas de ficheros distribuidos (no proporcionan E/S en paralelo) y los sistemas de ficheros paralelos (proporcionan una E/S de datos en paralelo).

Son ejemplos de sistema de archivos distribuidos: AFS, AppleShare, CIFS (también conocido como SMB o Samba), Coda, InterMezzo, NSS (para sistemas Novell Netware 5), NFS.

Son ejemplos de sistema de archivos paralelos: PVFS, PAFS.

Sistemas de archivos de propósito especial. Aquellos tipos de sistemas de archivos que no son ni sistemas de archivos de disco, ni sistemas de archivos de red.

Ejemplos: acme (Plan 9), archfs, cdfs, cfs, devfs, udev, ftpfs, lnfs, nntpfs, plumber (Plan 9), procfs, ROMFS, swap, sysfs, TMPFS, wikifs, LUFS, etc.

Comparativa

A continuación vamos a realizar una comparativa de varios sistemas de archivos más usados actualmente y que más se adaptan a los propósitos de esta tesis de Master.

Limites

Sistema de Ficheros	Longitud máxima del nombre de archivo	Longitud máxima de la ruta	Tamaño máximo del archivo	Tamaño máximo del volumen
ext3	255 bytes	Límites no definidos	16 GB to 2 TB	2 TB to 32 TB
ext4	256 bytes	Límites no definidos	16 GB to 16 TB	1 EB (pero las herramientas de usuario lo limitan a 16 TB)
Lustre	255 bytes	Límites no definidos	320 TB en ext4	2^{20} EB en ext4 (10 PB probados)
GFS	255 bytes	Límites no definidos	2 TB to 8 EB	2 TB to 8 EB
GFS2	255 bytes	Límites no definidos	2 TB to 8 EB	2 TB to 8 EB
OCFS	255 bytes	Límites no definidos	8 TB	8 TB
OCFS2	255 bytes	Límites no definidos	4 PB	4 PB

Tabla 2: Diferentes límites en los sistemas de ficheros estudiados

Características

Sistema de Ficheros	Hard links	Links simbólicos	Block journaling	Case-sensitive	Case-preserving	Log de cambios en ficheros	Snapshot	Encriptación	Se puede redimensionar el volumen
ext3	Si	Si	Si	Si	Si	No	No	No	Si offline
ext4	Si	Si	Si	Si	Si	No	No	No	Si offline
Lustre	Si	Si	Si	Si	Si	Si en 2.0	No	No	Si
GFS	Si	Si	Si	Si	Si	No	No	No	Desconocido
GFS2	Si	Si	Si	Si	Si	No	No	No	Si
OCFS	No	Si	No	Si	Si	No	No	No	Desconocido
OCFS2	Si	Si	Si	Si	Si	No	No	No	Si online. versión 1.4 y superiores

Tabla 3: Principales características de los sistemas de ficheros

Case-preserving: Cuando un sistema de ficheros guarda la información "Mayúsculas" sea o no Case-sensitive. Ejem.: NTFS no es case-sensitive pero si que es case-preserving.

Log de cambios en ficheros: Guarda los cambios en ficheros y directorios. Suele guardar información de creación, modificación, links, cambio de permisos, etc. Se diferencia del journaling porque este solo sirver para dos propositos generales: Backups, replicaciones, etc. Y auditorias del sistema de ficheros.

Sistemas Operativos Soportados

Sistema de Ficheros	Windows	Linux	Mac OS X	FreeBSD
ext3	Parcial con Ext2 IFS o ext2fsd	Si	con fuse-ext2 y ExtFS	Si
ext4	No	Si desde kernel 2.6.28	Parcial con fuse-ext2	No
Lustre	Parcial – en desarrollo	Si	Parcial - via FUSE	Parcial - via FUSE
GFS	Desconocido	Si	Desconocido	No
GFS2	No	Si	Desconocido	No
OCFS	Desconocido	Si	Desconocido	No
OCFS2	Desconocido	Si	Desconocido	No

Tabla 4: Sistemas Operativos que soportas estos sistemas de ficheros

Dado que mediante el estudio de las tablas anteriores, centrandonos en los sistemas de ficheros de cluster, no podemos apreciar grandes diferencias entre ellos, lo que nos indicará la mejor opción para nuestro sistema estará en función de los estudios de rendimiento que realizaremos en el capítulo 4 de este documento.

De los sistemas de archivos anteriores nos vamos a centrar en Lustre, GFS2 y OCFS2 dado que se tratan de sistemas de ficheros para clusters y se ajustan a las características que requiere la presente propuesta.

Lustre

Es un sistema de archivos distribuido Open Source, normalmente utilizado en clusters a gran escala. El nombre es una mezcla de Linux y clusters. El proyecto intenta proporcionar un sistema de archivos para clusters de decenas de miles de nodos con petabytes de capacidad de almacenamiento, sin comprometer la velocidad o la seguridad y está disponible bajo la GNU GPL.

Gracias a su gran rendimiento y escalabilidad, utilizar Lustre en sistemas MPP (*Massively Parallell Processor)* es lo más adecuado.

Global File System (GFS2)

El sistema de archivos GFS2 es un sistema de archivos nativo que interactúa directamente con la interfaz del sistema de archivos del kernel de Linux (VFS). Un sistema de archivos GFS2 puede ser implementado en un sistema independiente o como parte de una configuración de cluster. Cuando se implementa como un sistema de archivo de cluster, GFS2 emplea metadatos distribuidos y varios diarios.

GFS2 proporciona compartición de datos entre los nodos GFS2 de un cluster, con una única visualización consistente del espacio de nombres del sistema de archivos entre todos los nodos de GFS2. Esto le permite a los procesos en nodos diferentes compartir archivos GFS2 del mismo modo que los procesos en el mismo nodo pueden compartir archivos en un sistema de archivos local, sin ninguna diferencia discernible.

OCFS2

OCFS2 es un Sistema de ficheros en Cluster que permite el acceso simultáneo de multiples nodos. Cada nodo OCFS2 dispone de un sistema de ficheros montado, regularmente escribe un fichero meta-data permitiendo a los otros nodos saber que se encuentra disponible.

Virtualización

A continuación vamos a describir las diferentes tecnologías de virtualización analizándo con detalle las ventajas e inconvenientes de este tipo de sistemas, dando paso en último lugar a la descripción de varias de las herramientas de virtualización existentes en el mercado que más se adaptan a las características que estamos buscando en este estudio y que detallamos en la propuesta del mismo.

Hipervisor

Un **hipervisor** o **monitor de máquina virtual** es una plataforma que permite aplicar diversas técnicas de control de virtualización para utilizar, al mismo tiempo, diferentes sistemas operativos (sin modificar o modificados en el caso de paravirtualización) en un mismo equipo.

Los hipervisores pueden clasificarse en dos tipos:

- **Hipervisor tipo 1**: También denominado nativo. Software que se ejecuta directamente sobre el hardware, para ofrecer la funcionalidad descrita, ver figura 4.

Figura 4: Hipervisor nativo

Algunos de los hipervisores tipo 1 más conocidos son los siguientes: VMware ESXi, Xen, Citrix XenServer, Microsoft Hyper-V Server.

- **Hipervisor tipo 2**: También denominado hosted. Software que se ejecuta sobre un sistema operativo para ofrecer la funcionalidad descrita, ver figura 5.

Figura 5: Hipervisor hosted

Algunos de los hipervisores tipo 2 más utilizados son los siguientes: Oracle: VirtualBox,

VMware: Workstation y Server, Qemu, Microsoft: Virtual PC, Virtual Server.

Tecnología Intel VT-x o AMD-V de virtualización por Hardware

Desde que VMware desarrollara la virtualización para plataformas x86 en 1999, la virtualización por Hardware ha ido en constante evolución. Con esta tecnología, el hipervisor puede virtualizar eficientemente todo el conjunto de instrucciones x86 mediante la acción clásica de atrapar y emular el modelo de Hardware, en lugar de Software.

Intel ha introducido soporte de virtualización por Hardware en sus procesadores, “VT-x” o "Vanderpool". Con estas extensiones, un procesador opera en uno de los dos modos siguientes:

- Modo root: Su comportamiento es muy similar al modo de operación estándar (sin VT-x), y este es el contexto en el que se ejecuta un monitor de máquina virtual (VMM o hipervisor).
- Modo no root: (o contexto Guest) está diseñado para el funcionamiento de una máquina virtual.

Una novedad notable es que los cuatro niveles de privilegio (anillos) son compatibles con esta tecnología, por lo que el sistema Guest teóricamente puede ejecutarse en cualquiera de ellos. VT-x define la transición de modo root a modo no-root (y viceversa) y los llama "VM de entrada” y “VM de salida".

El equivalente de las instrucciones VT-x por parte de AMD se llama AMD-V o SVM. Además, éstas incluyen la característica de paginación anidada a partir de los procesadores Phenom y Opteron.

Pros y Contras de la Virtualización

Básicamente, lo que pretendemos conseguir cuando virtualizamos nuestros servidores son 3 cosas. En primer lugar optimización de recursos, seguido de una reducción de costes y para finalizar, una dependencia mínima del hardware en el que corren.

A continuación vamos a detallar que pros y contras nos podemos encontrar cuando virtualizamos.

- **Índices de utilización más altos:** Antes de la virtualización, los índices de utilización del servidor y almacenamiento en los centros de datos de la empresa rondaban por debajo del 50% (normalmente, lo más común era encontarnos con unos indices del 10% al 15%). Mediante la virtualización, las cargas de trabajo pueden ser encapsuladas y transferidas a los sistemas inactivos o sin uso.
- **Consolidación de Recursos:** La virtualización permite la consolidación de múltiples recursos de TI. Más allá de la consolidación de almacenamiento, la virtualización proporciona una oportunidad para consolidar la arquitectura de sistemas, infraestructura de aplicación, datos y base de datos, interfaces, redes, escritorios, e incluso procesos de negocios, resultando en ahorros de costo y mayor eficiencia.
- **Uso/coste menor energía:** La electricidad requerida para que funcionen los centros de datos se reduce al tener que mantener menos hardware en funcionamiento.
- **Ahorro de espacio:** La virtualización nos permite correr muchos sistemas virtuales en menos sistemas físicos.
- **Recuperación de desastre/continuidad:** La virtualización puede incrementar la disponibilidad ya que nos proporciona una menor dependencia del hardware y proporcionar nuevas opciones para la recuperación de desastre.
- **Mejora en los procesos de clonación y copia de sistemas:** Mayor facilidad para la creación de entornos de test que permiten poner en marcha nuevas aplicaciones sin impactar a la producción, agilizando el proceso de las pruebas.

- **Balanceo dinámico** de máquinas virtuales entre los servidores físicos que componen el pool de recursos, garantizando que cada máquina virtual ejecute en el servidor físico más adecuado y proporcionando un consumo de recursos homogéneo y óptimo en toda la infraestructura.

Por otro lado la virtualización de sistemas operativos también tiene algunos puntos débiles a destacar:

- **Rendimiento inferior:** Varios sistemas operativos virtualizados y ejecutados a la vez nunca alcanzarán las mismas cotas de rendimiento que si estuvieran directamente instalados sobre el hardware.
- **Limitaciones en el Hardware:** No es posible utilizar Hardware que no esté gestionado o soportado por el hipervisor.
- **Exceso de máquinas virtuales:** Como no hay que comprar Hardware, el número de máquinas y servidores virtuales se dispara. Aumentando el trabajo de administración, gestión de licencias y riesgos de seguridad.
- **Desaprovechamiento de recursos:** Crear máquinas virtuales innecesarias tiene un coste en ocupación de recursos, principalmente en espacio en disco, RAM y capacidad de proceso.
- **Centralización de las máquinas en un único servidor:** Una avería del servidor host de virtualización afecta a todas las máquinas virtuales alojadas en él. Con lo cual, hay que adoptar soluciones de alta disponibilidad y replicación para evitar caídas de servicio de múltiples servidores con una única avería.
- **Portabilidad limitada entre plataformas de virtualización:** Como cada producto de virtualización usa su propio sistema, no hay uniformidad o estandarización de formatos y la portabilidad entre plataformas está condicionada a la solución de virtualización adoptada. Elegir GNU/Linux, Mac OS X, Windows, u otros como anfitrión es una decisión importante.

Herramientas de Virtualización

A continuación se presentan las principales herramientas de virutalización que se ha considerado se adaptan más a las características buscadas y están teniendo una mejor aceptación y más uso en la actualidad. En primer lugar se describirá brevemente cada una de ellas para concluir con una comparativa que se muestra en la tabla 5.

KVM

Se trata de una herramienta de libre distribución, que emplea la técnica de virtualización completa, usando las extensiones de virtualización por hardware Intel VT o AMD, para crear VMs que ejecutan distribuciones de Linux. Además requiere una versión modificada de Qemu para completar el entorno virtual.

OpenVZ

Es un proyecto de código abierto basado en Virtuozzo (software comercial), utiliza una técnica de virtualización a nivel de sistema operativo y trabaja bajo distribuciones Linux, donde la compañía SWsoft ha puesto su código bajo la licencia GNU GPL. OpenVZ carece de las propiedades de Virtuozzo, pero ofrece un punto de partida para probarlo y modificarlo.

Virtual Box

Se distribuye bajo licencia GNU LGPL. También utiliza virtualización completa, dispone de una interfaz gráfica denominada Virtual Box Manage, que permite crear VMs con Windows o Linux y

su respectiva configuración de red. VirtualBox ofrece un mecanismo de acceso remoto a las VMs mediante RDP (Remote Desktop Protocol), protocolo desarrollado por Microsoft para acceder a escritorios remotos.

VMware

Esta herramienta también utiliza virtualización completa y la mayor parte de las instrucciones se ejecutan directamente sobre el hardware físico. Otros productos incluyen VMware Workstation que es de pago, los gratuitos VMware Server y VMware Player. VMware permite VMs con Windows y Linux.

Xen

Entorno de virtualización de código abierto desarrollado por la Universidad de Cambridge en el año 2003. Se distribuye bajo licencia GPL de GNU. Permite ejecutar múltiples instancias de sistemas operativos con todas sus características, pero carece de entorno gráfico. En el caso de requerirlo, se convierte en una herramienta de uso comercial.

Nombre	OS del Host	OS del Guest	Licencia	Soporte drivers OS guest	Metodo de Operación	Uso típico	Velocidad relativa al OS host	Soporte comercial
KVM	Linux	FreeBSD, Linux, Windows	GPL2	Si	AMD-V e Intel-VT-x	Servidor	Casi Nativo	RedHat o Novell
OpenVZ	Linux	Linux	GPL	Compatible	Virtualización a nivel de sistema operativo	Aislamiento de servidores virtualizados	Nativo	
VirtualBox	Windows, Linux, Mac OS X x86, FreeBSD	Linux, Mac OS X Server, FreeBSD,Windows	GPL2 y Versión comercial	Si	Virtualización	Servidor y Escritorio	Casi Nativo	Si (con licencia comercial)
VMware ESX Server 4.0 (vSphere)	No host OS	Windows, Linux, FreeBSD, virtual appliances	Propietaria	Si	Virtualización	Servidores, cloud computing	Muy cerca del Nativo	Si
VMware Server	Windows, Linux	Windows, Linux, FreeBSD, virtual appliances	Propietaria	Si	Virtualización	Servidor y Escritorio	Muy cerca del Nativo	Si
Xen	NetBSD, Linux, Solaris	FreeBSD, NetBSD, Linux, Windows XP & 2003 Server (needs vers. 3.0 and an Intel VT-x o AMD-V)	GPL	No necesita, a excepción del driver de red para NAT. Se necesita un kernel especial o nivel de abstracción hardware para el guest.	Para-virtualización y full-virtualización	Servidor y Escritorio	Muy cerca del Nativo. Perdida sustancial de rendimiento en sobrecargas de red y disco.	Si

Nombre	Soporta USB	GUI	Asignación de memoria en caliente	Aceleración 3D	Snapshots por VM	Snapshot de sistemas en funcionamiento	Migración en caliente	PCI passthrough
KVM	Si	Si	Si	Si (via AIGLX[1])	Si	Si	Si	Si
OpenVZ	Si	No	Por Hardware (No maneja swap)	No			Si	
VirtualBox	USB 1.1 (USB 2.0 versión comercial)	Si	Si	OpenGL 2.0, DirectX 3D	Yes branched	Si	Si	Solo Linux
VMware ESX Server 4.0 (vSphere)	Si	Si	Si	Si		Si	Si	Si
VMware Server	Si	Si	Si	No		Si	No	
Xen		Si	Si	Si (con VMGL)		Si	Si	Si

Tabla 5: Comparativa global plataformas virtualización

[1] (Accelerated Indirect GLX), es un proyecto iniciado por RedHat y la comunidad Fedora Linux para permitir aceleración indirecta GLX, capacidad de render en X.Org y drivers DRI.

Clusters de Alta Disponibilidad

Para conseguir redundancia y protección contra fallos en un sistema, la primera medida a tomar suele ser replicar sus componentes hardware más críticos. Por ejemplo discos duros, fuentes de alimentación, interfaces de red, etc.
Estas medidas aumentan el nivel de disponibilidad de un sistema, pero para conseguir un nivel aún más alto, se suelen utilizar configuraciones de hardware y software (clusters de Alta Disponibilidad).

Un *Cluster de Alta Disponibilidad* es un conjunto de dos o más servidores, que se caracteriza por compartir el sistema de almacenamiento yporque están constantemente monitorizándose entre sí. Si se produce un fallo de hardware o de los servicios de alguna de las maquinas que forman el cluster, el software de alta disponibilidad es capaz de rearrancar automáticamente los servicios que han fallado en cualquiera de los otros equipos del cluster. Y cuando el servidor que ha fallado se recupera, los servicios se migran de nuevo a la máquina original.

Además de para caidas de servicio no programadas, la utilización de clusters es útil en paradas de sistema programadas como puede ser un mantenimiento hardware o una actualización software.

En general las razones para implementar un cluster de alta disponibilidad son:

- Aumentar la disponibilidad
- Escalabilidad
- Tolerancia a fallos
- Reducción tiempos de recuperación ante fallos

Configuraciones de Alta Disponibilidad

Las configuraciones más comunes en este tipo de entornos son activo/activo y activo/pasivo.

Configuración Activo/Activo

En una configuración activo/activo, todos los servidores del cluster pueden ejecutar los mismos recursos simultáneamente. Es decir, todos los servidores poseen los mismos recursos y pueden acceder a estos independientemente de los otros servidores del cluster. Si un nodo del sistema falla y deja de estar disponible, sus recursos siguen estando accesibles a través de los otros servidores del cluster.

La ventaja principal de esta configuración es que los servidores en el cluster son máseficientes ya que pueden trabajar todos a la vez. Sin embargo, cuando uno de los servidores deja de estar accesible, su carga de trabajo pasa a los nodos restantes, lo que puede producir una sobrecarga del servidor que sigue en pie y por lo tanto una degradación en los servicios ofrecidos.

Configuración Activo/Pasivo

Un cluster de alta disponibilidad en una configuración activo/pasivo, consiste en un servidor que posee los recursos del cluster y otros servidores que son capaces de acceder a esos recursos, pero no los activan hasta que el propietario de los recursos ya no este disponible.

Las ventajas de la configuración activo/pasivo son que no hay degradación de servicio y que los servicios sólo se reinician cuando el servidor activo deja de responder. Sin embargo, una desventaja de esta configuración es que los servidores pasivos no proporcionan ningún tipo de recurso mientras están en espera, haciendo que la solución sea menos eficiente que el cluster de tipo activo/activo.

Funcionamiento de un cluster de alta disponibilidad

En un cluster de alta disponibilidad, el software de cluster realiza dos funciones fundamentales. Por un lado intercomunica entre sí todos los nodos, monitorizando continuamente su estado y detectando fallos. Y por otro lado administra los servicios ofrecidos por el cluster, teniendo la capacidad de migrar dichos servicios entre diferentes servidores físicos como respuesta a un fallo.

A continuación se describen los elementos y conceptos básicos en el funcionamiento del cluster.

Comunicación entre nodos

El software de cluster gestiona servicios y recursos en los nodos además de mantener continuamente entre estos una visión global de la configuración y estado del cluster. De esta forma, ante el fallo de un nodo, el resto conoce que servicios se deben restablecer.

Dado que la comunicación entre los nodos del cluster es crucial para el funcionamiento de este, es recomendable utilizar una conexión independiente, que no se pueda ver afectada por problemas de seguridad o rendimiento.

Heartbeat

El software de cluster conoce en todo momento la disponibilidad de los equipos físicos, gracias a la técnica de heartbeat. El funcionamiento es sencillo, cada nodo informa periódicamente de su existencia enviando al resto una “señal de vida”.

Escenario Split-Brain

El split-brain se produce cuando más de un servidor o aplicación pertenecientes a un mismo cluster intentan acceder a los mismos recursos, lo que puede causar daños a dichos recursos. Este escenario ocurre cuando cada servidor en el cluster cree que los otros servidores han fallado e intenta activar y utilizar dichos recursos.

Monitorización de Recursos (Resource Manitoring)

Ciertas soluciones de clustering HA permiten no sólo monitorizar si un host físico esta disponible, también pueden realizar seguimientos a nivel de recursos o servicios y detectar el fallo de estos.

Reiniciar Recursos

Cuando un recurso falla, la primera medida que toman las soluciones de cluster es intentar reiniciar dicho recurso en el mismo nodo. Lo que supone detener una aplicación o liberar un recurso y posteriormente volverlo a activar.

Migración de Recursos (Failover)

Cuando un nodo ya no está disponible, o cuando un recurso fallido no se puede reiniciar satisfactoriamente en un nodo, el software de cluster reacciona migrando el recurso o grupo de recursos a otro nodo disponible en el cluster.

Dependencia entre recursos

Los recursos y servicios del cluster se pueden agrupar según necesidades y/o dependencia entre ellos, obligando al cluster a gestionar las acciones pertinenetes sobre todos ellos simultáneamente.

Preferencia de Nodos (Resource Stickiness)

Podemos encontrarnos en casos en los que ciertos servicios o recursos debamos ejecutarlos en un cierto nodo del cluster o por cualquier motivo sea más interesante priorizar la ejecución de dicho recurso o servicio en unos nodos u otros. Para ello se pueden establecer preferencias que gestionen estas prioridades.

Fencing

En los clusters HA existe una situación donde un nodo deja de funcionar correctamente pero todavía sigue levantado, accediendo a ciertos recursos y respondiendo peticiones. Para evitar que el nodo corrompa recursos o responda con peticiones, los clusters lo solucionan utilizando una técnica llamada Fencing.

La función principal del Fencing es hacerle saber a dicho nodo que esta funcionando en mal estado, retirarle sus recursos asignados para que los atiendan otros nodos y dejarlo en un estado inactivo.

Quorum

La comunicación dentro de un cluster puede sufir problemas que pueden derivar en situaciones de Split-Brain. Para evitar estas situaciones se puede introducir un sistema de votaciones para evaluar la situación de cada nodo conjuntamente por la mayoría de nodos disponibles y así poder levantar los servicios y recursos en los nodos con mayoría y dejar inactivos los que esten en minoría.

Soluciones Open Source de Clustering HA

Existen muchos proyectos Open Source dedicados a proporcionar soluciones para Clusters de Alta Disponibilidad en Linux, y teniendo en cuenta que actualmente las aplicaciones de clustering son bastante complejas, suelen constar de varios componentes, por lo que solemos encontrarnos en situaciones en las que una solución completa de clustering utiliza componentes de varios subproyectos.

A continuación vamos a describir algunos proyectos y componentes más importantes en la actualidad dentro en el ámbito de clusters de Software Libre.

Proyecto Linux-HA y Heartbeat

El proyecto Linux-HA [HA004] tiene como objetivo proporcionar una solución de alta disponibilidad (clustering) para Linux.

Linux-HA se utiliza ampliamente y como una parte muy importante en muchas soluciones de Alta Disponibilidad. Desde que comenzó en el año 1999 a la actualidad, sigue siendo una de las mejores soluciones de software HA para muchas plataformas.

El componente principal de Linux-HA es Heartbeat, un demonio que proporciona los servicios de infraestructura del cluster (comunicación y membresía).

Para formar una solución cluster de utilidad, Heartbeat necesita combinarse con un Cluster Resource Manager (CRM), que realiza las tareas de iniciar o parar los recursos y dotar de alta disponibilidad.

En la primera versión de Linux-HA, se utiliza con Heartbeat un sencillo CRM que sólo erá capaz de administrar clusters de 2 nodos y detectar fallos a nivel de maquina. Con Linux-HA 2 se desarrolló un nuevo CRM más avanzado, que superaba dichas limitaciones. De este nuevo desarrollo surge el proyecto CRM Pacemaker.

Pacemaker CRM

El proyecto Pacemaker [HA001] surge en el año 2007, a raíz de la segunda generación de Linux-HA. Los programadores del componente CRM de Linux-HA, deciden extraer el desarrollo y mantenimiento de éste en un proyecto separado. Para que el nuevo CRM pueda utilizar como capa de comunicación no sólo Heartbeat si no también OpenAIS.

Pacemaker es compatible totalmente con Heartbeat, así como con los scripts de recursos existentes para este, también se ha adaptado el administrador gráfico Linux-HA para que funcione con Pacemaker.

Pacemaker esta disponible en la mayoría de las distribuciones Linux actuales, las cuales lo han adoptado como sucesor de Heartbeat.

OpenAIS

OpenAIS Cluster Framework [HA006] es una implementación open source de las Application Interface Specification (AIS). Un conjunto de especificaciones para estandarizar el desarrollo de servicios e interfaces para la alta disponibilidad, desarrolladas por el Service Availability Forum [HA005].

Los principales beneficios de una solución de Cluster HA basado en las normas AIS son la mejora en portabilidad e integración, permite sistemas más escalables, la reducción de costes y reutilización de componentes.

Esta estandarización puede ser muy beneficiosa no sólo para los componentes principales del software o middleware de clustering, si no por el hecho de que el cluster sea capaz de monitorizar un mayor número de servicios y recursos con un API unificada.

El proyecto OpenAIS implementa actualmente los componentes de infraestructura y membresía. Y es utilizado en soluciones completas de clustering como Pacemaker o RedHat Cluster.

RedHat Cluster Suite

RedHat-Cluster Suite [HA002] es un proyecto de desarrollo open source de diferentes componentes de clustering para Linux. Dicho proyecto se basa casi en la totalidad del producto RedHat Cluster Suite para su distribución comercial Linux RHLE.

RedHat-Cluster es un conjunto de componentes que forman una solución de clustering HA completa y que utiliza un CRM propio llamado CMAN.

La arquitectura se está basada en el uso de OpenAIS como componente de mensaje/membresía y CMAN como administrador de recursos (CRM). Así como otros componentes que proporcionan fencing, balanceo de carga, o las propias herramientas de administración del cluster.

RedHat-Cluster también esta disponible para otras distribuciones Linux que no sean RedHat.

Corosync Cluster Engine

Corosync Cluster Engine [HA007] es un proyecto open source bajo la licencia BSD, derivado del proyecto OpenAIS. El objetivo principal del proyecto es desarrollar una solución de cluster completa, certificada por la OSI (Open Source Initiative), con soporte para Linux, Solaris, BSD y MacOSX.

El proyecto se inicia en Julio de 2008 y la primera versión estable Corosync 1.0.0 se lanzó en agosto de 2009.

Otros

Existen otros muchos proyectos dedicados a facilitar la instalación y configuración de clusters de alta disponibilidad. Por ejemplo el proyecto UltraMonkey, que combina LVS + Heartbeat + Ldirector, para proporcionar una solución de cluster HA y balanceo de carga.

Así como otros proyectos de clusters de alta disponibilidad completos que han quedado descatalogados con los años, como el caso de Kimberlite o de OpenHA.

También hay varios proyectos muy interesantes para plataformas diferentes a Linux, como el caso del Open High Availability Cluster (OHAC) que es la versión OpenSource del Solaris Cluster de Sun Microsystems.

Soluciones comerciales

Dentro del ámbito empresarial, las compañías RedHat y Novell, ofrecen soluciones completas de clusters de alta disponibilidad basadas en los proyectos libres mencionados anteriormente.

Estos paquetes comerciales se venden como una solución completa de software libre más soporte anual, documentación y actualizaciones de seguridad.

Además, podemos encontranos con soluciones libres de XenServer y VMWare VSphere pero que para obtener gran parte de las funcionalidades avanzadas dentro de un cluster, vamos a tener que optar por las versiones de pago de estas mismas aplicaciones.

Suites de gestión de recursos de clusters

Para realizar una evaluación y comparativa práctica de lo que nos pueden ofrecer estas herramientas hemos preparado dos servidores dedicados para crear un cluster mínimo que proporcione alta disponibilidad.

Los servidores dispondrán de una tarjeta de red conectada a internet y otra con un cable cruzado. Cada máquina contará con recursos, programas, librerías y sistemas operativos independientes.

Los servidores tienen la siguiente configuración:

Quad core

4GB RAM

Disco Sistema 500GB 7.2k. sataII

Disco Datos 500GB 7.2k. sataII

2x Tarjeta Intel 1Gb/s

Las herramientas de clustering seleccionadas deberán cumplir los siguientes requisitos:

- El servidor físico (host) tendrá instalado un sistema operativo Linux (distribuciones Ubuntu, CentOS y/o Fedora)
- Deberá soportar DRBD .
- Se necesita acceso mínimo para la configuración del servidor (soporte de video, mouse y teclado) .
- Deberá soportar sistemas de ficheros de disco compartido.
- Deberá soportar migración en caliente de máquinas virtuales.

Entre los requisitos opcionales estarán:

- Soporte para fence
- Soporte para DLM
- Herramienta gráfica para su gestión

Comparativa

Para este estudio se han realizado las instalaciones de estas dos herramientas en los servidores anteriormente detallados, realizando pruebas de funcionamiento y observando el funcionamiento de las características que dicen sus especificaciones que ofrecen, que detallamos a continuación.

RedHat Cluster Suite

La RedHat Cluster Suite es un conjunto de componentes para crear clusters de alta disponibilidad y balanceo de carga. Las herramientas que la componen son las siguientes [HA003]:

- **CCS:** cluster configuration system para gestionar el fichero de configuración del cluster
- **CLVM:** extensión de **LVM2** para su uso en clusters
- **CMAN:** cluster manager
- **DLM:** distributed lock manager: gestión de bloqueos
- **Fence:** Sistema para evitar accesos no deseados al disco. Puede implementarse deshabilitando el puerto de fibra hacia el sistema de storage (**SAN**) o resetando la maquina en cuestión
- **GFS:** sistema de ficheros de disco compartido (**Global File System**)
- **GFS2:** segunda versión del sistema de ficheros de disco compartido (**Global File System 2**)
- **GNBD:** módulo para proveer de acceso directo a los discos a través de la red
- **GULM:** sistema de bloqueo y clustering (alternativo a **CMAN** con **DLM**)
- **OpenAIS:** infraestructura de cluster
- **Magma:** capa para facilitar la transición entre **GULM** y **CMAN/DLM**
- **RGManager:** resource group manager: se encarga de verificar y gestionar el funcionamiento de los servicios y sus recursos.
- **Conga:** es un conjunto integrado de componentes de software que proporciona tareas de configuración, administración centralizada para los cluster y el almacenamiento de Red Hat. Conga ofrece las siguientes funcionalidades:
  - Interfaz de web para administrar cluster y almacenaje
  - Implementación automatizada de los datos del cluster y paquetes de soporte
  - Integración fácil con los cluster existentes
  - No hay necesidad de reautenticación
  - Integración de los registros y estado del cluster
  - Control detallado sobre los permisos de usuarios

Pacemaker

Pacemaker es un administrador de recursos del clúster orientado a obtener la máxima disponibilidad de sus recursos mediante la detección, recuperación de nodos y los fallos a nivel de recursos, haciendo uso de las capacidades de mensajería y la pertenencia a la infraestructura de cluster proporcionada por corosync, heartbeat o openais.

Entre las principales características destacan:

- Detección y recuperación de errores de nodo y a nivel de servicio
- Independiente de almacenamiento, sin necesidad de almacenamiento compartido
- Independiente de recursos

- Soport STONITH para garantizar la integridad de datos
- Soporta grandes y pequeños clusters
- Soporta prácticamente cualquier configuración de redundancia
- La actualizacion de la configuración se replica automáticamente desde cualquier nodo
- Soporta tanto quórum y agrupaciones de recursos

Característica/herramienta	**RedHat Cluster Suite**	**Pacemaker**
Tipo de clusters	Alta disponibilidad y balanceo de carga (LVS)	Alta disponibilidad
Herramienta de configuración	CMAN (texto), Conga (gráfica - web)	CRM (texto)
Gestión de la configuración del cluster	CCS	CIB
DLM	Si	Si
Fence	Si	Si
Almacenamiento compartido	SAN, iSCSI, FC	SANs, ISCSI, FC, cLVM, DRBD
Sistemas de Ficheros de cluster	GFS,GFS2,NFS,CIFS,CLVM	GFS2,OCFS2
Infraestructura	OpenAIS	OpenAIS,Corosync
Monitorización	RGManager	crm
Nodos	Más de 128	Máximo no establecido. Se puede usar tanto en clusters grandes como pequeños
Coste	Gratuito	Gratuito

Tabla 6: Comparativa Suites de gestión de clusters

Conclusiones

Como soluciones de alta disponibilidad hemos descartado VMWare VSphere y XenServer por tratarse de soluciones de pago. Aunque existen versiones libres, para obtener funcionalidades avanzadas como la migración en caliente y fence entre otras se tiene que recurrir a las versiones de pago.

De las soluciones presentadas, las dos pueden ser utilizadas de manera óptima en diferentes entornos. Considero que la elección entre una u otra solución puede ir en función de:

- Sistema de ficheros de cluster a usar.
- Escalabilidad (si se pretende montar un cluster de mas de 128 nodos).
- Entorno de configuración. Aunque existen herramientas de configuración gráficas para Pacemaker, son herramientas de terceros y no vienen integradas con la propia suite del cluster.

Pueden existir alternativas que en otras circunstancias sean las más recomendadas, sobre todo si no atendemos a la variable del coste, ya que VMWare VSphere y XenServer son unas grandes, buenas y estables soluciones para alta disponibilidad de servidores virtualizados, pero su coste puede llegar a ser muy elevado.

Capitulo 3

Propuesta para alta disponibilidad y flexiblidad con bajo coste

Uno de los principales objetivos del presente trabajo es conseguir un sistema que nos permita ofrecer alta disponibilidad en servidores y un mejor aprovechamiento de recursos hardware a bajo coste, para que cualquier empresa y/o servicio, ya sea con pequeño o mediano presupuesto, pueda disponer de este tipo de configuraciones y dotar a sus servicios de una mejor disponibilidad, una menor dependencia del hardware del que dispongan y un mejor aprovechamiento del mismo gracias a la virtualización.

No obstante, cuando hablamos de alta disponibilidad en servidores debemos tener presente que no es lo mismo que intentar ofrecer alta disponibilidad en servicios. En primer lugar, los tiempos para migrar un servicio caido en un servidor a otro nodo del cluster suelen ser inferiores a los tiempos para migrar un servidor completo, y por otro lado, si estamos monitorizando servidores, no tenemos porque darnos cuenta si un servicio que esta corriendo en un determinado servidor esta fallando o no, con lo que podriamos tener un servicio no disponible durante un tiempo indeterminado hasta que otras herramientas externas al cluster nos informasen de que estos servicios están fallando.

Además, tal como mostramos en las figuras 7 y 8, pretendemos encontar un sistema que nos aporta alta disponibilidad en los servicios que ofrecemos (aplicanto técnicas de clusters de alta disponibilidad), sin tener que perder la flexibilidad de poder usar estos sobre los sitemas operativos que mejor se adapten a sus características (unión de alta disponibilidad y virtualización de servidores) y optimizando al máximo los recursos de los que disponemos para poder reducir costes en hardware y consumos.

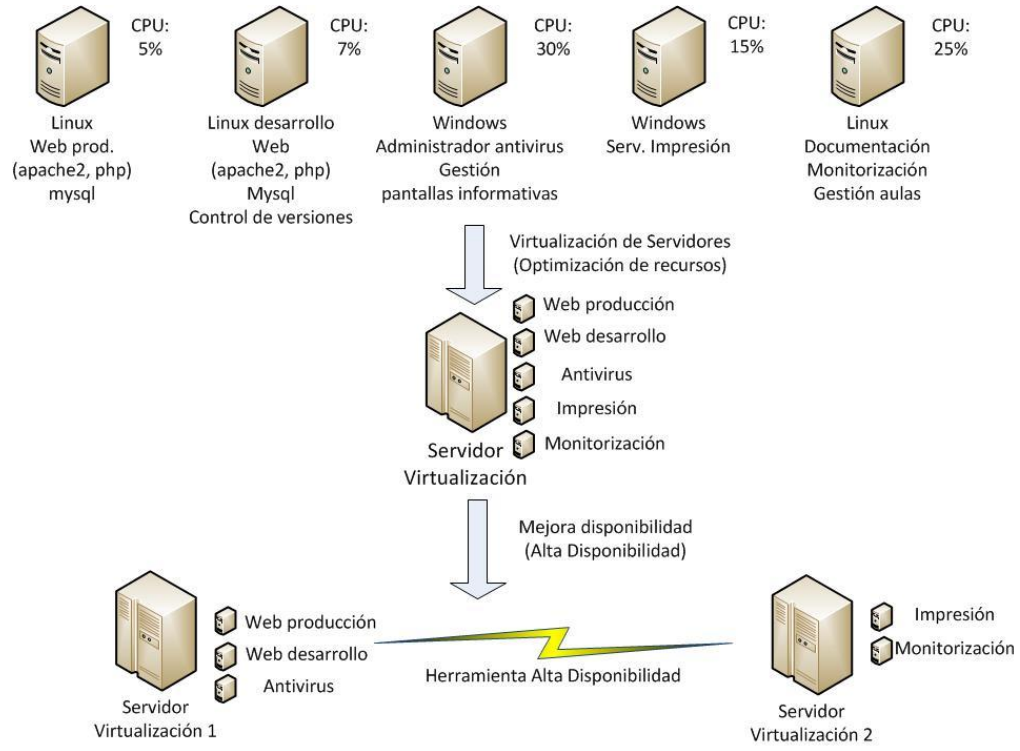

Figura 7: Unión virtualización de servidores y alta disponibilidad

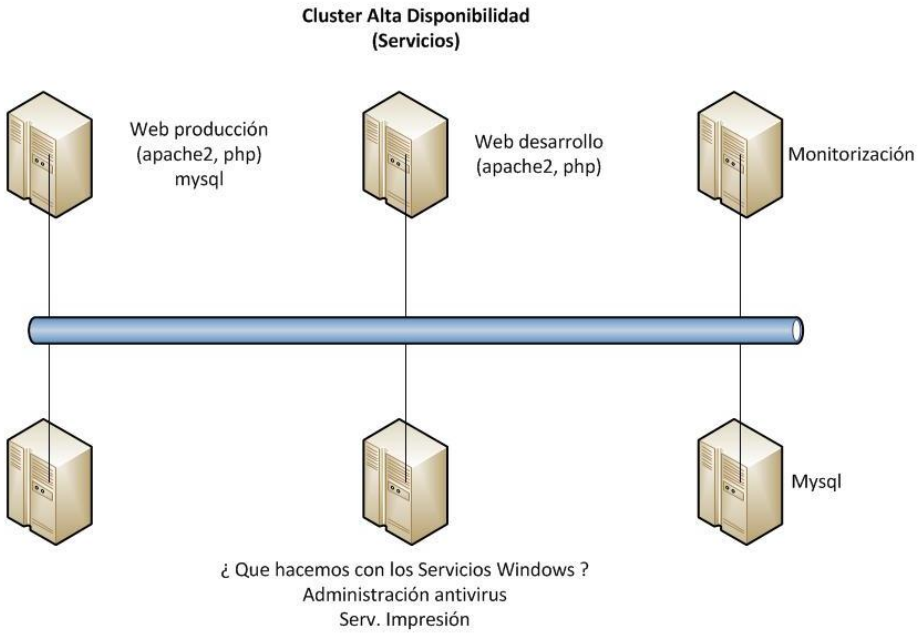

Figura 8: Cluster de alta disponibilidad para servicios

Por otro lado, debemos tener en cuenta que los servidores van a necesitar siempre (virtualizados o no) sus paradas de mantenimiento (reinicios por actualizaciones, instalaciones, etc.), además de seguir sufriendo las paradas no previstas a causas de fallos del software y/o configuraciones, y a todo esto le tendremos que añadir la dependencia de los sistemas de virtualización y los servidores host donde se hospedan y sus posibles fallos. Con lo cual, al intentar dotar de una menor dependencia del hardware y de la posibilidad de tener los servidores corriendo ante cualquier fallo de los servidores físicos donde se hospedan, ***les introducimos unos puntos de fallos que actualmente suelen ser más probables que los fallos hardware y que son el fallo del sistema host y el software de virtualización.*** Pero eso sí, adicionalmente y para paliar estos fallos, dotamos al sistema de una solución de alta disponibilidad para que tanto los fallos hardware como los software de los sistemas host nos permita tener una repercusión mínima en el servicio.

Concretando, vamos a dotar a nuestro sistema de una mejora en el MTTR (Mean Time To Repear) reduciéndolo sustancialmente, a costa de decrementar un poco el MTTR (Mean Time To Failure) y el MTBF (Mean Time Between Failures) ya que incorporamos al sistema posibilidad de fallos adicionales que antes no teniamos, tales como el sistema operativo host y la plataforma de virtualización.

Como alternativa a las observaciones anteriores, mediante combinaciones de estas técnicas, podríamos llegar a configurar clusters de alta disponibilidad para nuestros servicios, basandonos en servidores virtualizados, con lo que podriamos a llegar a reducir (mejorar) los tiempos de indisponibilidad (como describiamos anteriormente, la migración de servicios suele ser menos costosa en terminos de tiempo que la migración de servidores).

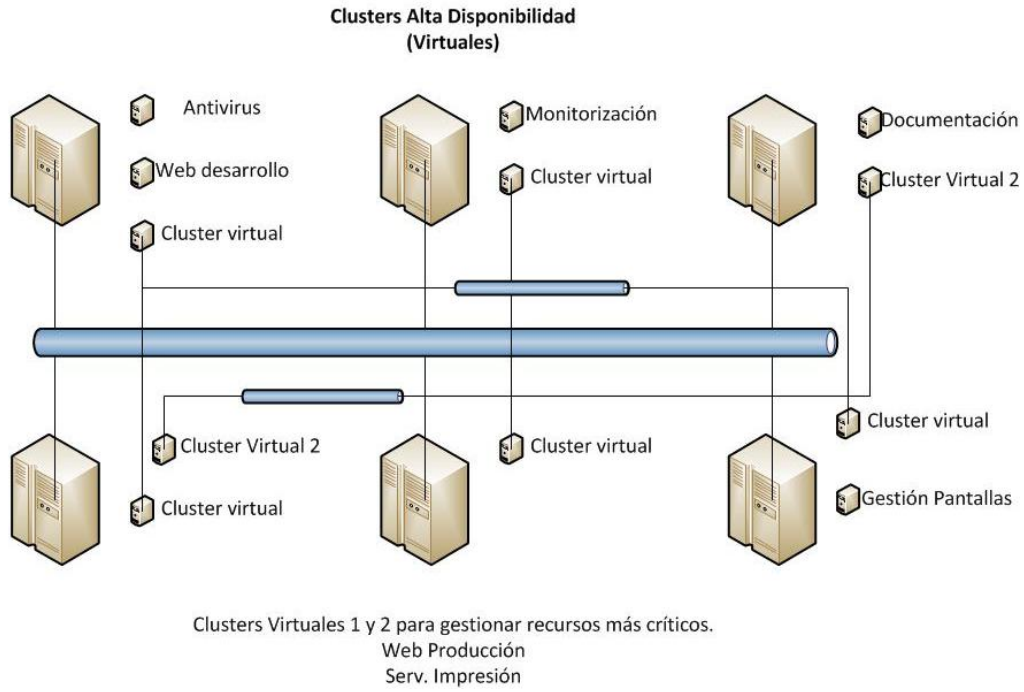

Figura 9: Cluster de alta disponibilidad ejecutandose sobre servidores virtualizados

En nuestra propuesta, siguiendo la filosofía de obtener un sistema de bajo coste, reaprovechando al máximos los recursos que tenemos y optimizando su uso. Vamos a centrar y concretar la configuración en un cluster mínimo de 2 servidores, si bien la propuesta es generalizable, a cualquier número de servidores así como otros parámetros, todo ello en función de los requisitos finales del mismo.

Estos dos servidores que formarán el cluster mínimo propuesto comparten sistema de almacenamiento, ejecutan simultáneamente servidores virtualizados y entre ellos se monitorizan mediante las soluciones de cluster seleccionadas tras el estudio previo.

A continuación, vamos a detallar las diferentes propuestas de cada uno de los principales componentes del sistema que pretendemos conseguir para poder llegar a una configuración óptima y que evalúaremos posteriormente.

Sistema almacenamiento

En el entorno que pretendemos montar, el principal cuello de botella viene de la mano de los sistemas de almacenamiento, ya que al virtualizar, entre otras cosas, para aprovechar mejor los recursos hardware de los que disponemos vamos ha hacer que sobre unos mismos discos se ejecuten simultaneamente varios servidores que antes se ejecutaban en sistemas de almacenamiento independientes. Por lo tanto, vamos a dedicar un gran esfuerzo a estudiar esta parte del entorno para intentar encontrar tanto el mejor sistema de almacenamiento como el mejor sistema de ficheros que se adapte a las premisas de esta tesis de Master, en el capítulo 4 se exponen dichos estudios así como sus resultados y conclusiones.

Actualmente no podemos negar que el uso de dispositivos SAN para dotar a nuestros sistemas de almacenamiento compartido es una muy buena opción, eso sí, siempre que dispongamos de un buen presupuesto.

Como en nuestro caso el presupuesto es uno de los componentes que intentamos minimizar, nos

vamos a centrar en opciones mucho más económicas y que nos aporte, a poder ser, las mismas prestaciones.

Para ello haremos uso de tecnologías como NFS y DRBD, las cuales vamos a comparar en el apartado de experimentación.

Virtualización

La reducción en el MTTR va a depender de la posibilidad que tengamos de independizar nuestros servidores virtualizados al máximo y de las máquinas físicas donde se estén ejecutando.

Actualmente existen multitud de plataformas de virtualización que nos aportan en mayor o menor medida las necesidades que estamos buscando.

En un primer paso vamos a descartar todas aquellas que sea preciso pagar licencia por su uso y que no dispongan de versión gratuita. Vamos a descartar también aquellas que no nos permitan realizar full virtualization o para-virtualización. Y vamos a escoger entre las que cumplan los requisitos anteriores, las que disponen actualmente de un desarrollo y soporte más activo por parte de la comunidad.

- VMWare Server
- Sun VirtualBox
- Xen
- KVM

Cluster

Para poder reducir al máximo el MTTR, además de la plataforma de virtualización comentada anteriormente, vamos a necesitar una solución de cluster de alta disponibilidad que nos permita monitorizar y migrar los servidores virtualizados de un sistema host a otro en el menor tiempo y la mejor manera posible.

En el estudio de las diferentes soluciones de clusters de alta disponibilidad nos vamos a centrar en el estudio de Pacemaker y RadHat Cluster Suite, ya que cumplen con los siguentes factores que consideramos de mucho interés para nuestra solución final:

- Proyecto estable, con un desarrollo y soporte continuo por parte de la comunidad.
- Estandarización e interoperabilidad con otros proyectos o componentes de cluster.
- Libertad, disponibilidad y coste del software.
- Soporte en múltiples distribuciones gratuitas y comerciales.
- Proporciona disponibilidad de clase empresarial.
- Amplia compatibilidad con hardware y plataformas .
- Compatibilidad con el mayor número de servicios y recursos existentes.
- Escalabilidad de la solución.
- Compatibilidad con virtualización.
- Documentación existente y curva de aprendizaje del producto.

Capitulo 4

Experimentación

En esta sección se presenta un estudio práctico de las dos piezas fundamentales para lograr nuestro primer y principal objetivo, en el que intentamos reducir lo máximo posible la dependencia de nuestros servidores del hardware en el que corren, lo cual redunda en una mejor disponibilidad de los mismos. Para ello vamos a estudiar en primer lugar las distintas opciones de almacenamiento que se adapten a nuestras premisas y que será la base para ejecutar las imagenes de nuestros servidores virtualizados en las plataformas estudiadas en segundo lugar.

Estudio de prestaciones en sistemas de almacenamiento en clusters

El principal objetivo del presente estudio es evaluar las prestaciones de los sistemas de almacenamiento en clusters. Dado que es una parte fundamental de los sistemas basados en cluster y que en nuestra propuesta toma mayor relevancia debido a que sobre este sistema van a almacenarse las imagenes de los servidores virtualizados que van a corren en el mismo y de su rendimiento va a depender los servidores virtualizados que podamos ejecutar y el rendimiento de los mismos. Por lo tanto va a ser importante estudiar a fondo el rendimiento de cada uno de estos sistemas de almacenamiento y los sistemas de ficheros a usar.

Para ello se ha utilizado el entorno y metodología que se exponenen en los siguientes apartados.

Entorno

El escenario utilizado para las pruebas de almacenamiento consiste en dos equipos con la siguiente configuración:

2 x Quad core

16GB RAM

Disco Sistema 2 x 500GB en RAID1 HW. 7.2k sataII

Discos Datos 4 x 500GB en RAID5 HW. 7.2k sataII

Discos Datos 4 x 500GB en RAID5 HW. 15K7 sas

Controladora RAID SAS/SATA LSI SAS2108

Tarjeta Intel Ethernet Server Adapter X520-LR1 E10G41BFLR Fibra Monomodo 10Gb/s

y otros dos con la siguiente configuración:

Quad core

4GB RAM

Disco Sistema 500GB 7.2k. sataII

Disco Datos 500GB 7.2k. sataII

Tarjeta Intel 1Gb/s

Para las pruebas del sistema NFS, usaremos un NAS NSS4000 4-Bay Gigabit de Cisco con 4 discos de 500GB 7.2k en RAID5.

Las pruebas las realizamos sobre Ubuntu Server 10.04 LTS, con kernel 2.6.39, DRBD 8.3.10,

OCFS2 y GFS2. Las pruebas del sistema de ficheros lustre no las podemos realizar porque no existen paquetes disponibles para esta distribución y el código fuente encontrado no funciona en esta versión de kernel tal como se detalla en [SF001].

Herramientas de evaluación

Para el estudio de prestaciones entre los diferentes sistemas de almacenamiento propuestos y sus sistemas de ficheros vamos a hacer uso de la herramienta Iozone (ver anexo I).

Iozone es una herramienta de benchmark destinada a comprobar el rendimiento de un sistema de archivos. La aplicación genera y mide una gran cantidad de operaciones sobre ficheros y nos permitirá comparar el rendimiento combinado de cada uno de los sistemas de ficheros propuestos sobre sus sistemas de almacenamiento.

Para la obtención de datos ejecutamos el siguiente comando en cada unos de los equipos y sistemas de archivos a analizar.

Para los equipos con 4GB de RAM:

```
# iozone -R -s 128K -s 512K -s 1M -s 64M -s 128M -s 512M -s 1G -s 2G -s 3G -s 4G
-s 5G -i 0 -i 1 -i 2 -i 6 -i 7 -b lab.wks
```

Para los equipos con 16GB de RAM:

```
# iozone -R -s 128K -s 512K -s 1M -s 64M -s 128M -s 512M -s 1G -s 2G -s 5G -s
10G -s 18G -i 0 -i 1 -i 2 -i 6 -i 7 -b lab.wks
```

De esta forma obtendremos el comportamiento cuando utilizamos la memoria caché del procesador (tamaño del archivo inferior a la caché del procesador), cuando empleamos la memoria RAM (tamaño del archivo comprendido entre la caché del procesador y la cantidad total de memoria RAM) y cuando hacemos uso directamente de las operaciones de entrada/salida a disco (tamaño del archivo superior a la memoria RAM).

Hay que tener en cuenta que tras ejecutar cada una de estas pruebas es importante limpiar los datos cacheados en la memoria RAM.

A partir del kernel 2.6.16 se introdujo la posibilidad de actuar sobre la información cacheada en la memoria RAM. Para ello se tiene un fichero denominado **/proc/sys/vm/drop_caches** que es utilizado por el kernel para saber qué debe hacer con esos datos en función del número que se escriba en dicho fichero:

- 0: No hacer nada.
- 1: Liberar la *pagecache*.
- 2: Liberar *inodos* y *dentries*.
- 3: Liberar *pagecache*, *inodos* y *dentries*.

La pagecache es la memoria caché de páginas. Los inodes es la representación que utiliza el sistema operativo para los ficheros y directorios guardados en memoria o disco (un inodo contiene todos los metadatos necesarios para administrar objetos en el sistema de ficheros, incluyendo las operaciones que sean posibles sobre él). Y por último, los dentries se utilizan para mantener la relación entre los directorios y los ficheros, así como para realizar la traducción entre los nombres y los inodos.

Por lo tanto, si quisiéramos liberar la pagecache, los inodos y los dentries ejecutaríamos el siguiente comando:

```
# sync ; echo 3 > /proc/sys/vm/drop_caches
```

Se recomienda ejecutar previamente el comando *sync* para forzar la grabación en la caché de aquellos datos que pudieran estar pendientes.

Resultados y discusión

Para el análisis y discusión en este apartado hemos seleccionado una representación de las gráficas tanto en escritura como en lectura para los sistemas de ficheros de cluster ocfs2 y gfs2, así como el almacenamiento en red nfs y el sistema de ficheros local ext4 para tener un punto inicial de comparación. En el anexo II del presente documento se pueden consultar todos los datos obtenidos de los test realizados y las gráficas correspondientes.

También mostramos las gráficas obtenidas para rendimiento de lectura y escritura en el otro entorno de pruebas donde disponemos de discos SATAII y SAS, para poder comprobar el efecto de usar discos estándares SATAII de 7200 rpm frente a discos SAS de 15700 rpm

Graficas Rendimiento Escritura. En sistema con discos SATAII a 7k2 rpm.

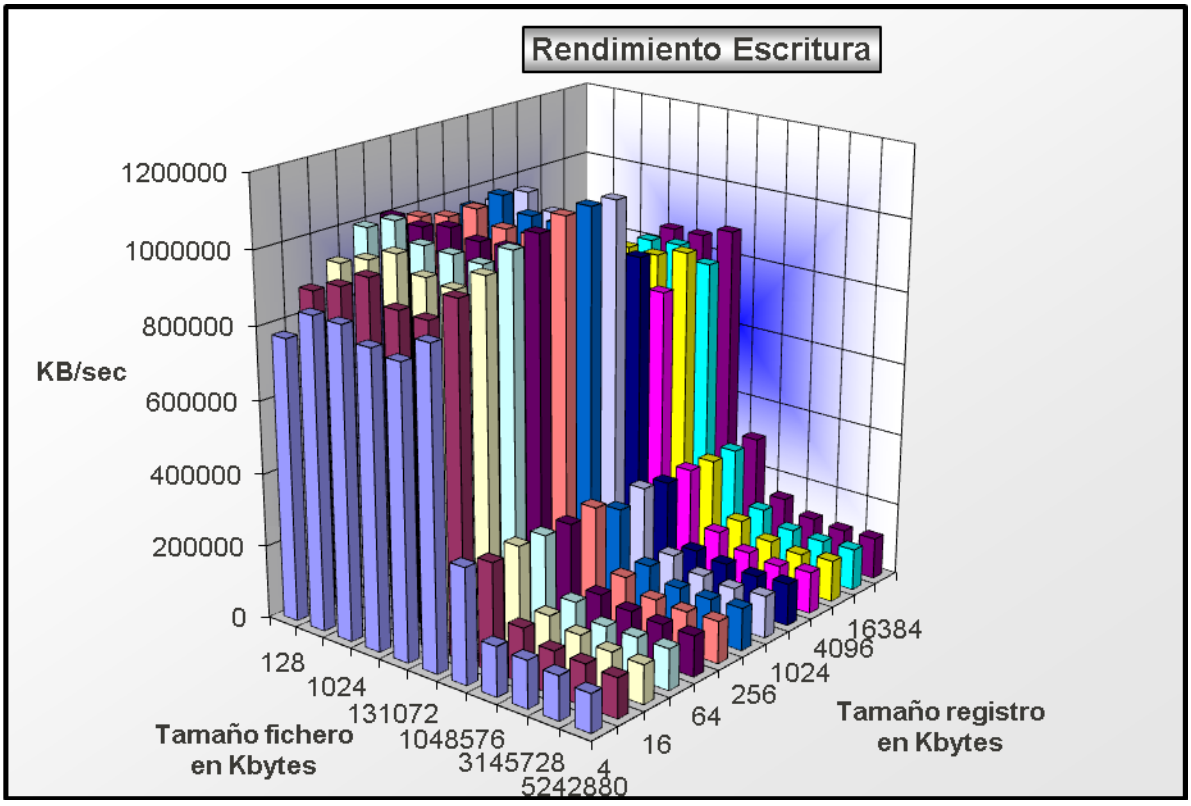

Figura 10: Sistema de Ficheros ext4 (Escritura)

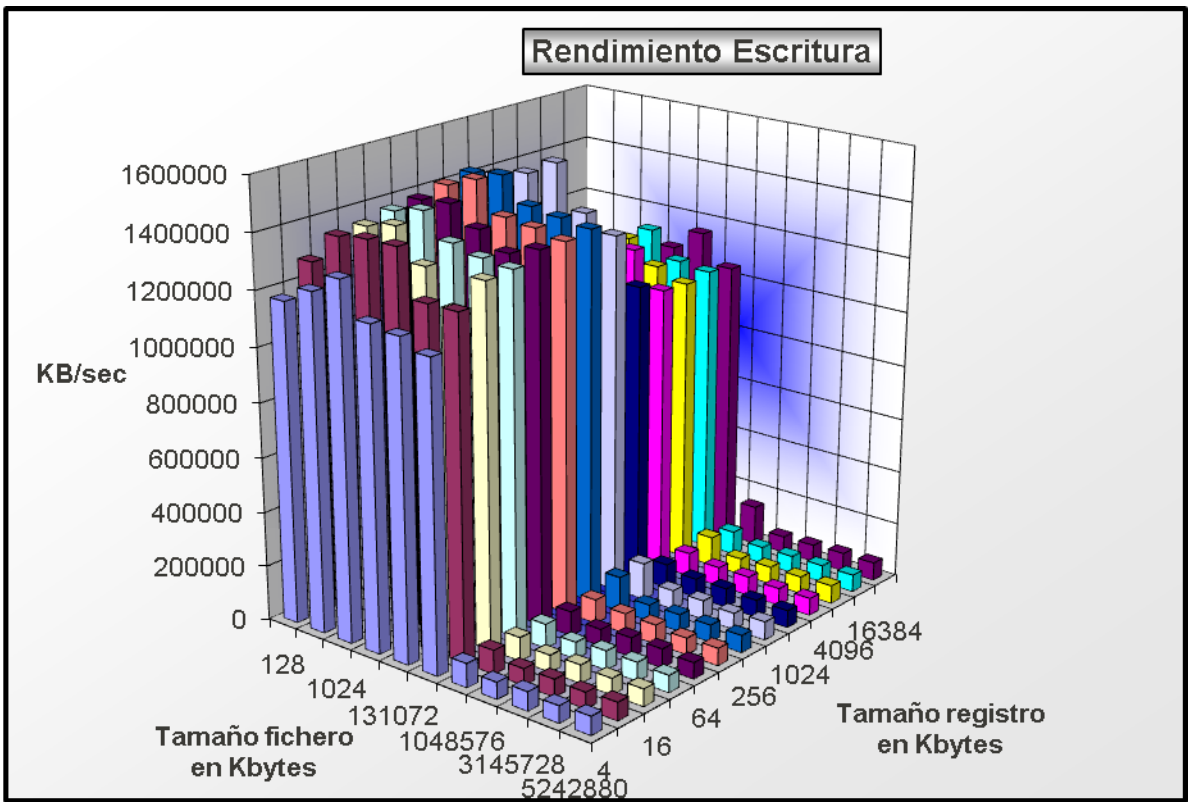

Figura 11: NFS - Sistema Ficheros ext4 (Escritura)

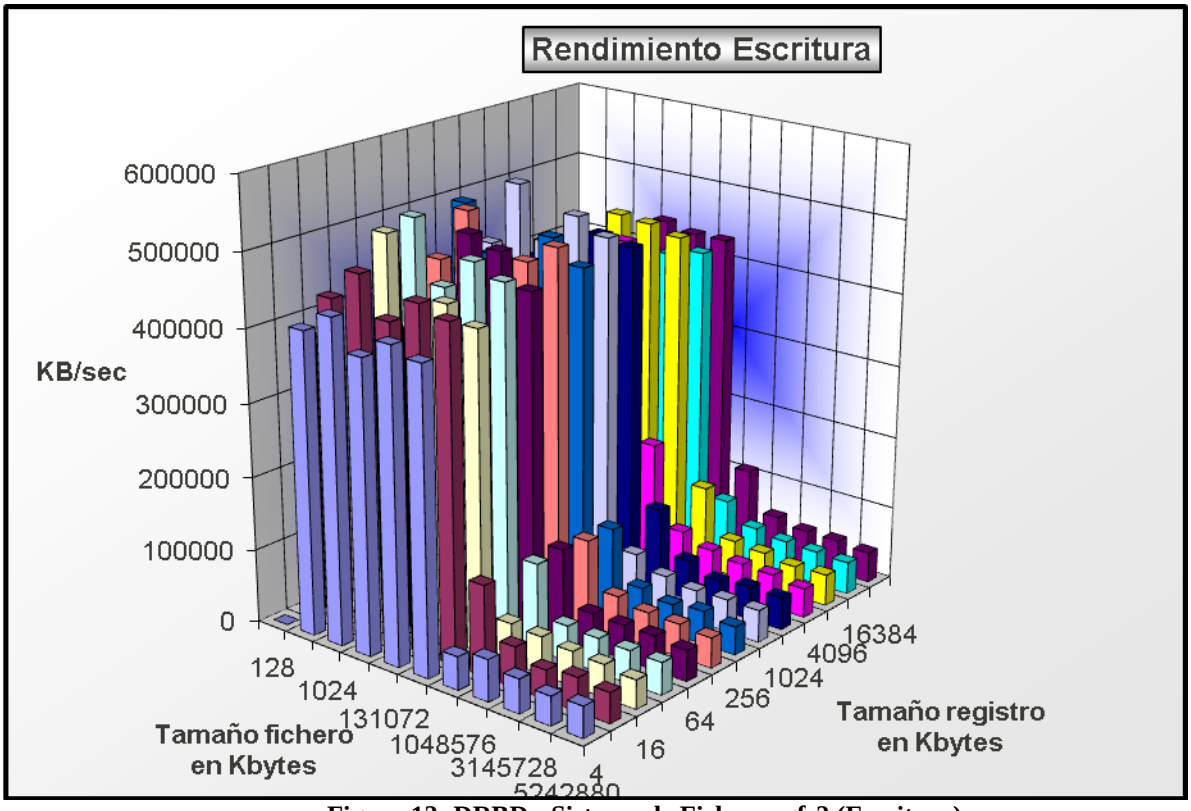

Figura 12: DRBD - Sistema de Ficheros gfs2 (Escritura)

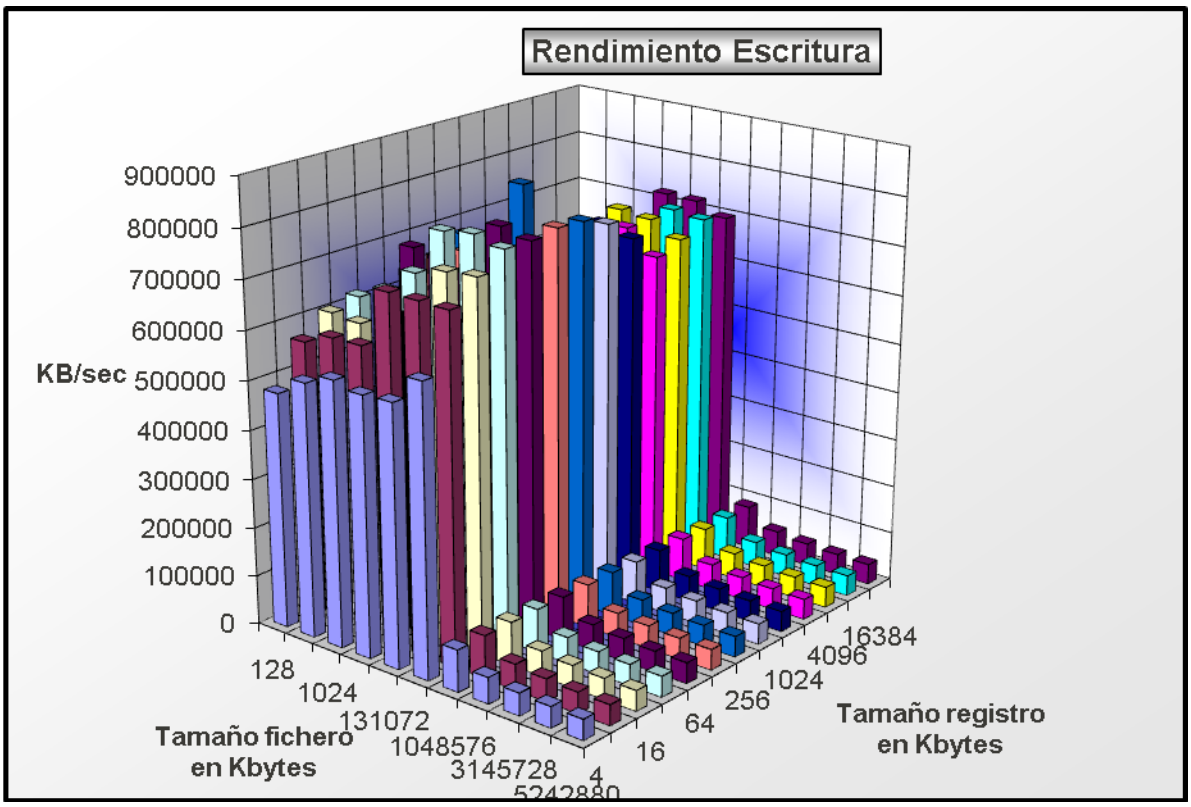

Figura 13: DRBD - Sistema de Ficheros ocfs2 (Escritura)

Graficas Rendimiento Lectura. En sistema con discos SATAII a 7k2 rpm.

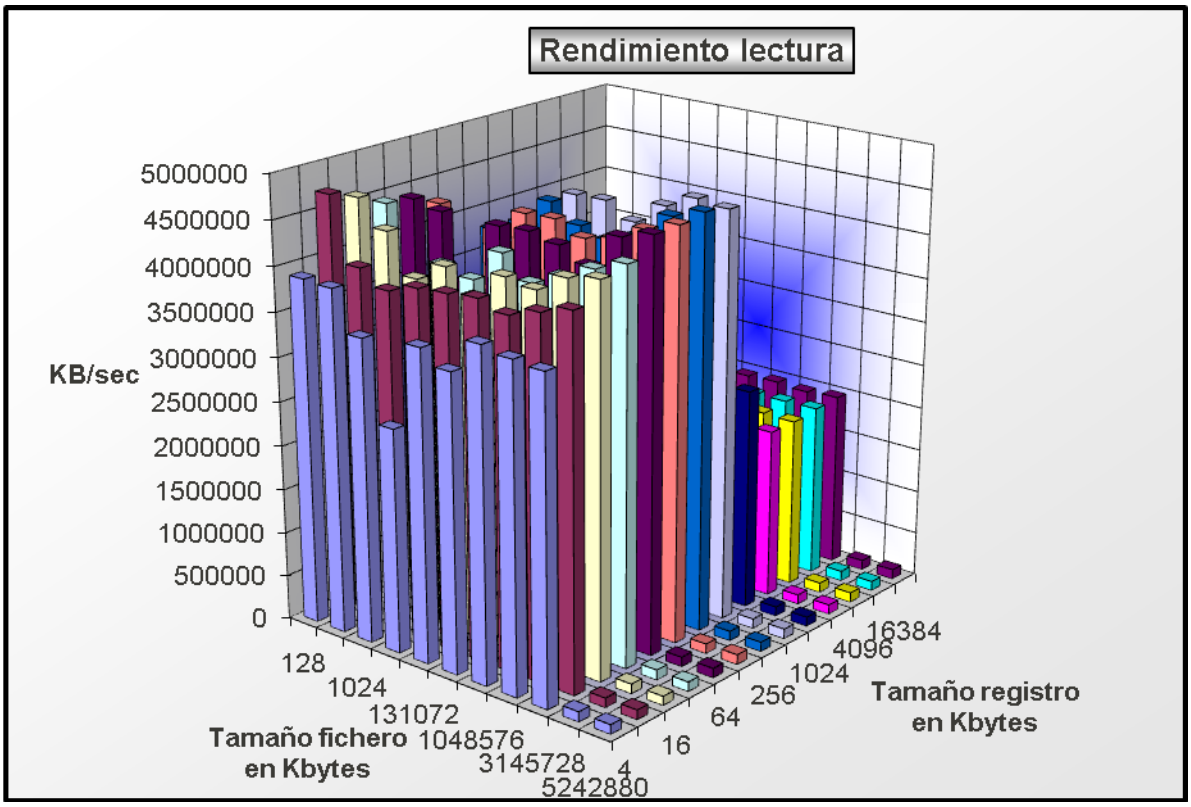

Figura 14: Sistema de Ficheros ext4 (Lectura)

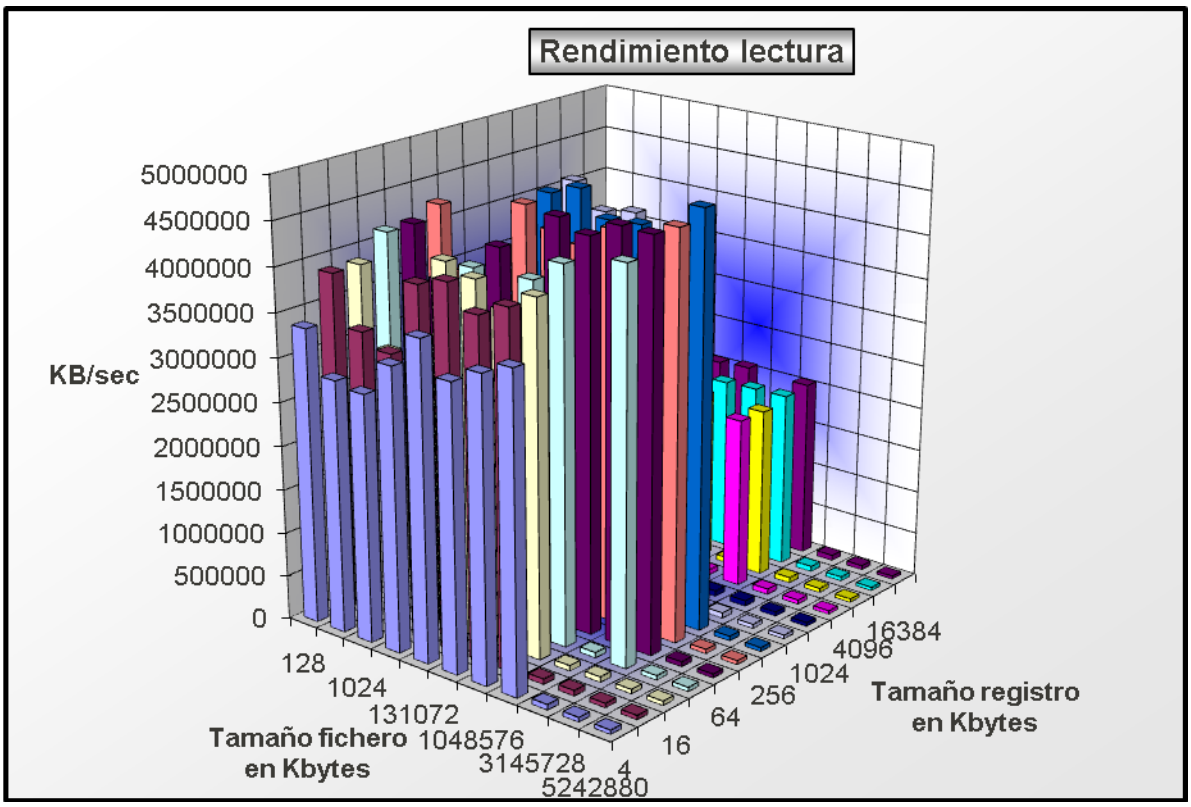

Figura 15: NFS - Sistema de Ficheros ext4 (Lectura)

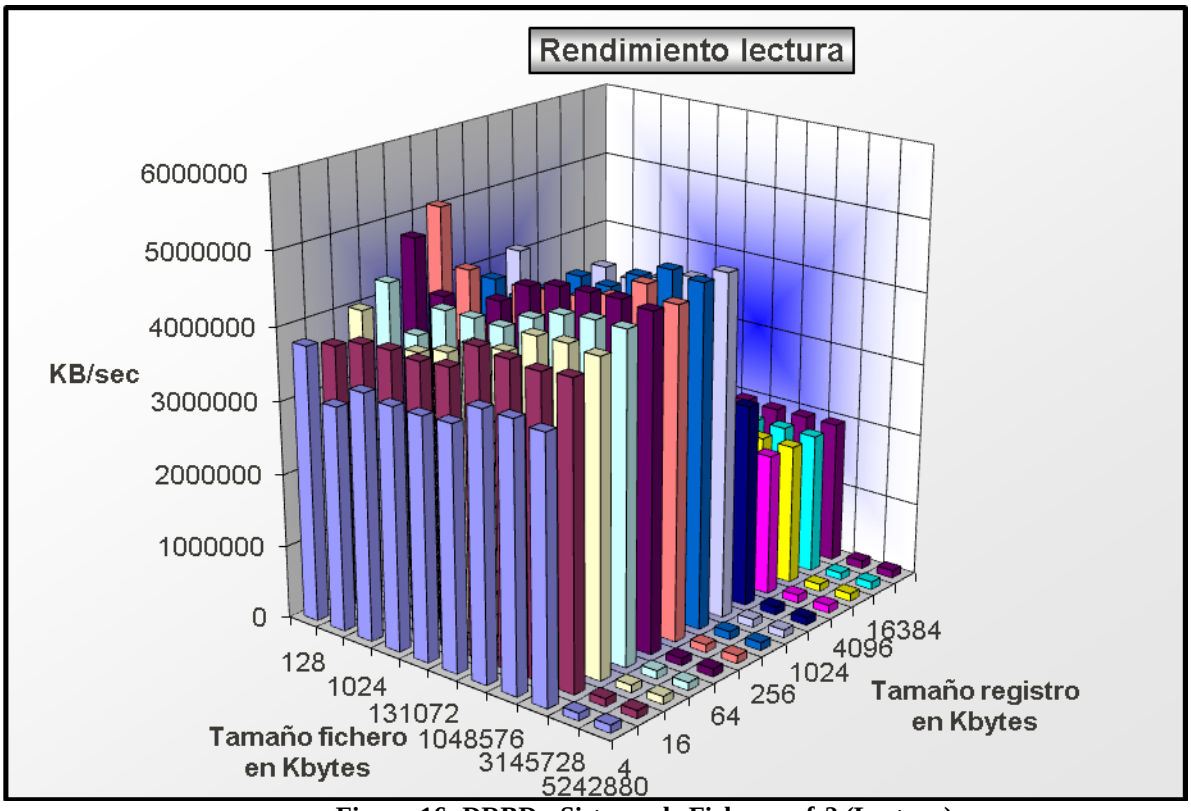

Figura 16: DRBD - Sistema de Ficheros gfs2 (Lectura)

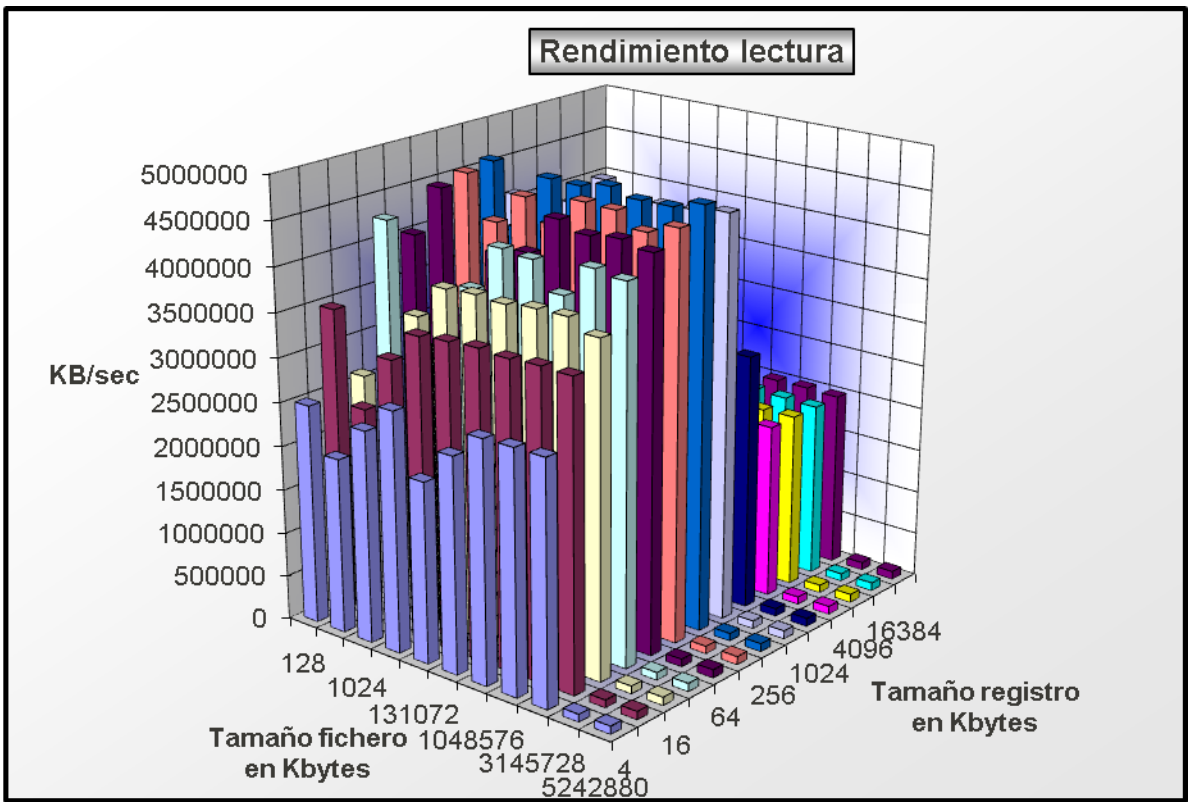

Figura 17: DRBD - Sistema de Ficheros ocfs2 (Lectura)

Graficas Rendimiento Escritura. En sistema con discos SATAII a 7k2 rpm y discos SAS 15k7 rpm.

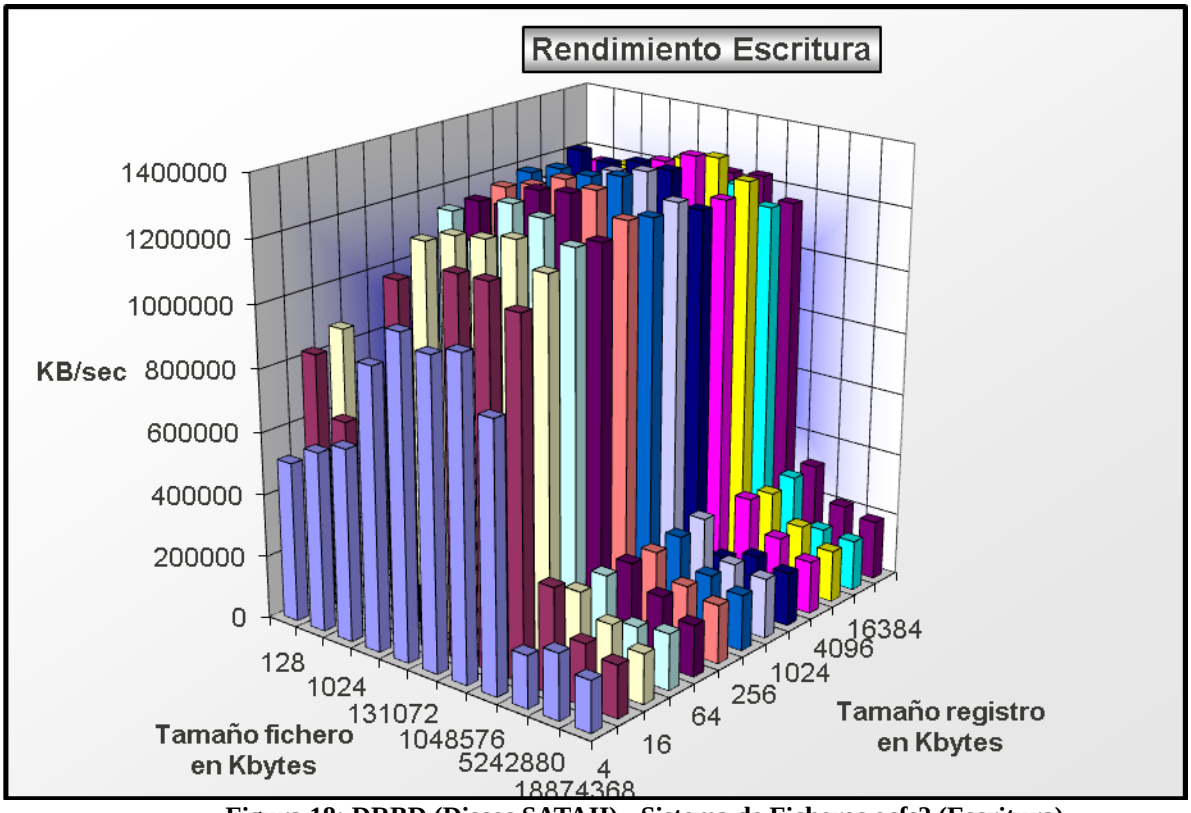

Figura 18: DRBD (Discos SATAII) - Sistema de Ficheros ocfs2 (Escritura)

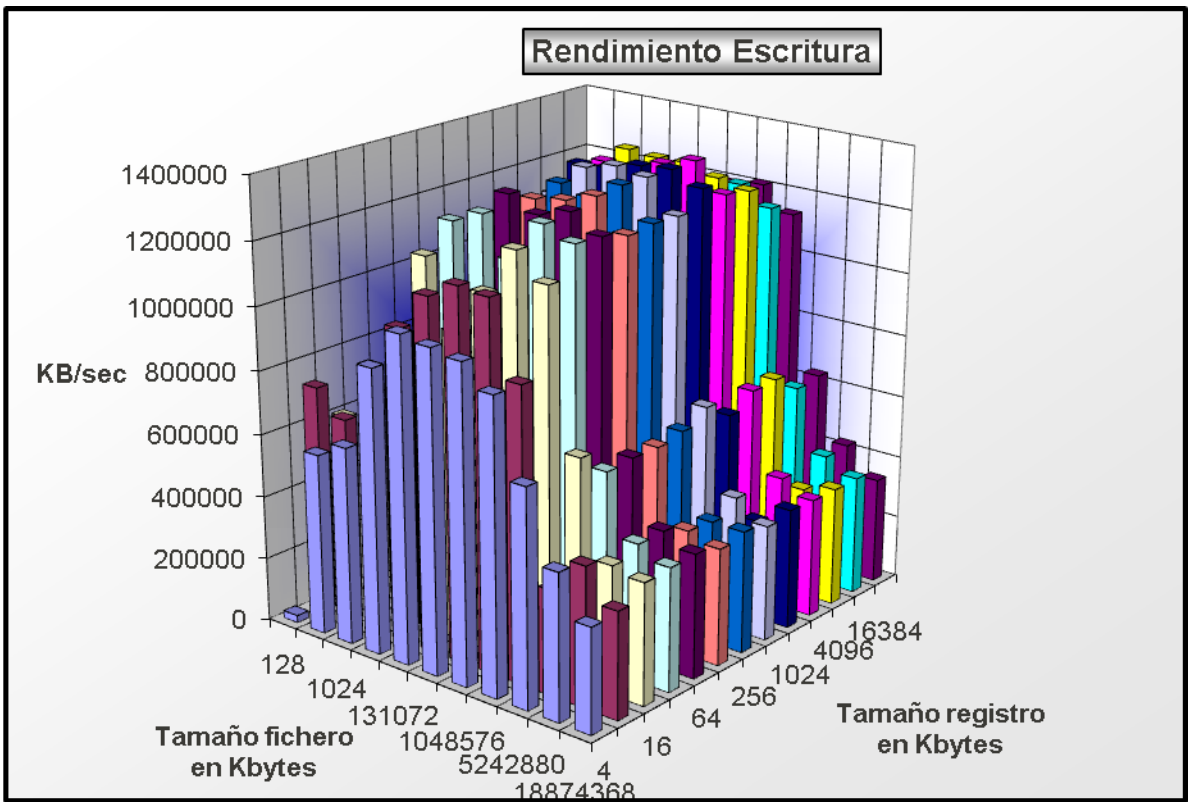

Figura 19: DRBD (Discos SAS) - Sistema de Ficheros ocfs2 (Escritura)

Graficas Rendimiento Lectura. En sistema con discos SATAII a 7k2 rpm y discos SAS 15k7 rpm.

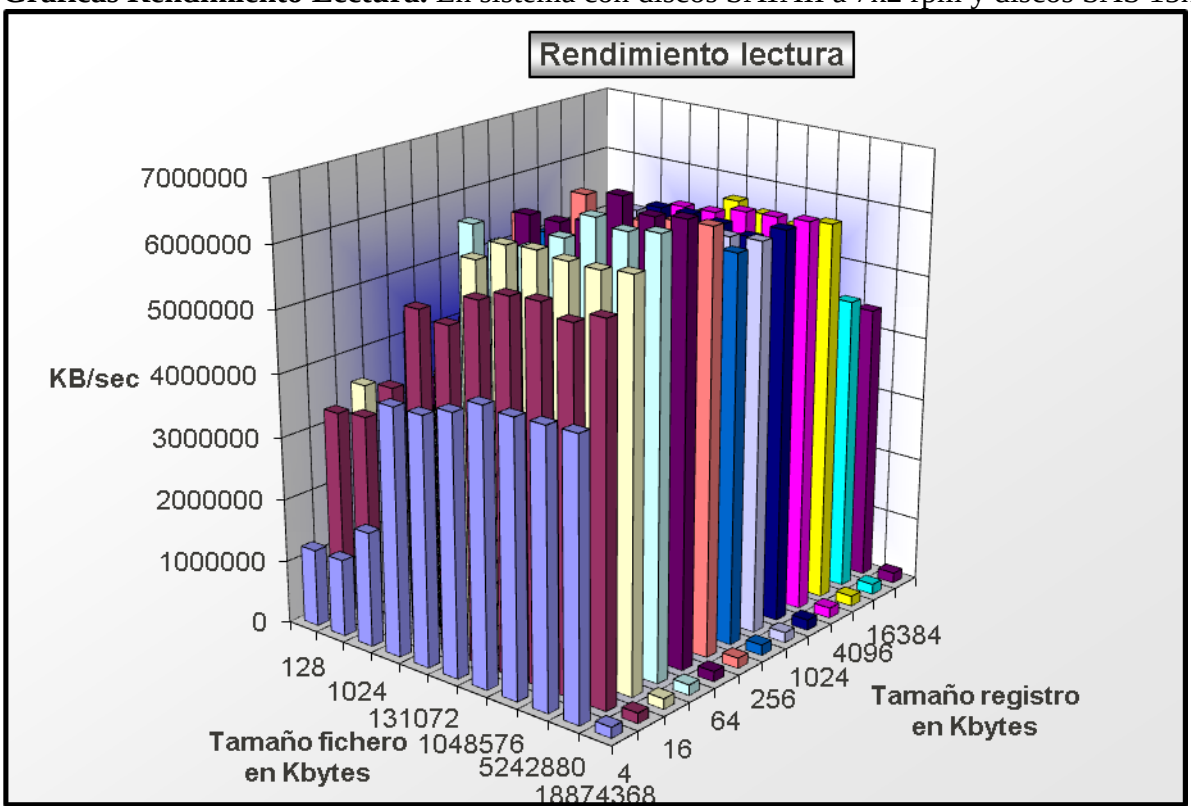

Figura 20: DRBD (Discos SATAII) - Sistema de Ficheros ocfs2 (Lectura)

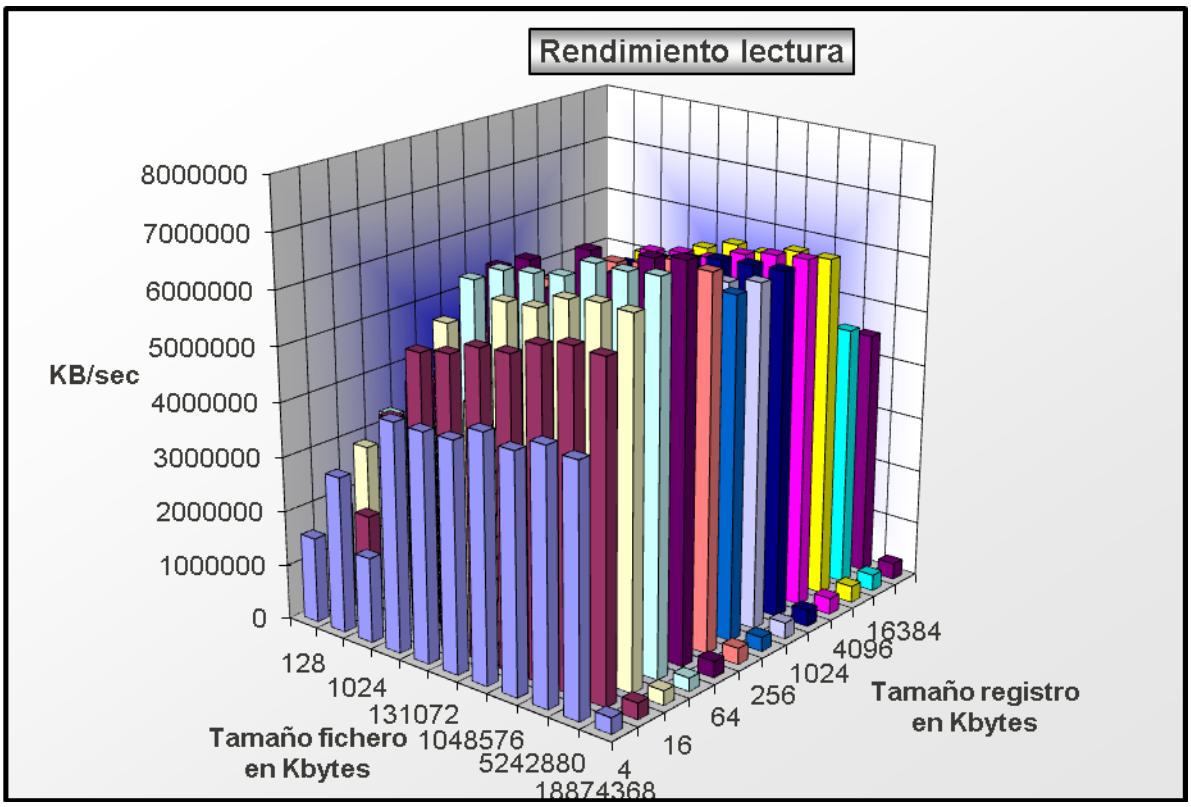

Figura 21: DRBD (Discos SAS) - Sistema de Ficheros ocfs2 (Lectura)

Conclusiones

Al estudiar los datos obtenidos en primer lugar podemos destacar, tal como era esperado, que el uso de discos SAS a 15500 rpm frente a los discos estándares SATAII de 7200 rpm nos aportan una gran diferencia en cuanto a rendimiento. No obstante, teniendo en cuenta que cada uno de los discos SAS que hemos usado en estas pruebas cuesta alrededor de 400€ frente a los 100€ que suelen costar los discos SATAII de estas capacidades, aún pensando en que una de las premisas de este documento es el montar un sistema de bajo coste, recomendamos el uso de estos discos SAS ya que no sólo nos permitirá correr con más agilidad las máquinas virtuales que necesitemos sino que podremos ejecutar más máquinas virtuales sobre los mismos sistemas. Ya que tal como hemos comentando varias veces a lo largo del documento, el principal cuello de botella de este tipo de sistema es el sistema de almacenamiento.

Por otro lado, si comparamos el uso de sistema NFS (almacenamiento en red) frente al DRBD propuesto, hemos encontrado dos ventajas que nos hacen decidirnos por este último. En primer lugar, el uso de DRBD mejora el rendimiento frente al NFS. En segundo lugar, con DRBD tendremos la posibilidad de tener nuestros datos duplicados en la red y separados geográficamente. No obtstante, el uso del DRBD aporta la desventaja de que solamente vamos ha poder acceder a los datos desde 2 nodos del cluster, frente a NFS que nos permitirá acceder desde muchos más nodos.

Para finalizar, si comparamos los sistemas de ficheros de cluster propuestos (gfs2 y ocfs2), podemos apreciar que en cuanto a rendimiento son muy parecidos, aunque ocfs2 ofrece mejores respuestas en escritura de archivos pequeños donde se beneficia mucho de la caché del sistema y gfs2 mejora bastante sus resultados en la lectura. No obstante lo que nos hace decidirnos por el uso de ocfs2 es su facilidad, comodidad de configuración y uso a la hora de poder ejecutar dicho sistema de ficheros en cada uno de los nodos del cluster o independientemente en un nodo aislado, cosa que nos facilitará la recuperación de datos en caso de desastre, frente a gfs2 que complica mucho el acceso a los datos en caso de tener que realizar una recuperación desde un nodo aislado.

Virtualización

Metodología de evaluación

A la hora de evaluar una plataforma de virtualización, nos vamos a centrar en el consumo de CPU y memoria que tiene cada una de las seleccionadas, así como el rendimiento que tienen estas en el momento del arranque de los sistemas operativos instalados en ellas.

No obstante, los parámetros que más interesan a nuestro fin, vienen dados por el uso continuado e intensivo de cada una de estas plataformas, entre otros, nos interesan conocer datos como la estabilidad, la facilidad de administración, el consumo de recursos sobre el host físico, las posibilidades de gestionar las máquinas en caliente (migración y cambio de parámetros) y por supuesto el coste de las licencias y/o soporte de la plataforma.

Entorno

Se preparan dos servidores dedicados que ejecute máquinas virtuales independientes entre sí, con el propósito de realizar las pruebas de software pertinentes.

Los servidores dispondrán de una tarjeta de red conectada a internet y otra con un cable cruzado para sincronizar los discos configurados mediante DRBD entre los dos nodos, para poder realizar pruebas de migración de las máquina virtuales en caliente entre los servidores. Cada máquina contará con recursos, programas, librerías y sistemas operativos independientes.

Los servidores tienen la siguiente configuración

Quad core

4GB RAM

Disco Sistema 500GB 7.2k. sataII

Disco Datos 500GB 7.2k. sataII

2x Tarjeta Intel 1Gb/s

La plataforma de virtualización deberá cumplir los siguientes requisitos:

- El servidor físico (host) tendrá instalado un sistema operativo Linux (distribuciones Ubuntu, CentOS y/o Fedora)
- Crearemos máquinas virtuales (guests) con los sistemas operativos Linux, Solaris/OpenSolaris, Windows XP, Windows Vista, Windows 2000-2003 Server y Windows 2008
- Se necesita acceso mínimo para la configuración del servidor (soporte de video, mouse y teclado)
- También debe contar con conexión de red (preferentemente de tipo bridge)

Entre los requisitos opcionales estarán:

- Soporte de interfaz de usuario fluido (para uso como Desktop)
- Soporte de acceso a recursos del sistema de manera directa (raw)
- Integración de la interfaz de usuario (portapapeles, cambio de resolución)
- Capacidad de modificar recursos sin necesidad de reiniciar la máquina virtual (‘en caliente’)

Evaluación y resultados

De entre la diversidad de alternativas tecnológicas que existen, hemos delimitado las pruebas a realizar a las siguientes soluciones por ser las más conocidad/usadas y las que mejor se adaptan al planteamiento de esta tesis de Master:

- VMWare Server
- Sun VirtualBox
- Xen
- KVM

Para evaluar estas herramientas en primer lugar vamos a realizar un estudio comparativo de las características de cada una de ellas. A continuación realizaremos una serie de test en los que mediremos el rendimiento de las mismas comparando los valores de consumo de CPU y memoria RAM, asi como la efectividad con la que gestiona el sistema de ficheros.

VMWare

Sitio web: http://www.vmware.com/

- VMware es la solución más conocida y con mayor presencia comercial, además de ofrecer según muchos administradores un excelente servicio de soporte por parte de la compañía.
- Una particularidad de VMWare Server es que la interfaz de configuración y consola es accesible vía una interfaz Web. La consola es una extensión disponible para Firefox.
- Los drivers adicionales (vmware-tools) tanto para Windows como para Linux mejoran notablemente la integración de la consola y en menor medida la performance de los discos.
- El controlador o driver escogido para los discos virtuales (IDE, SATA, SCSI, etc.) impacta de manera notable en el desempeño de la máquina virtual.
- Al instalar VMWare sobre ciertos sistemas operativos, a veces es necesario parchear el instalador de vmware-server para ponerlo en funcionamiento (dependiendo sobre todo de la versión del kernel usada).
- En nuestras pruebas, la ejecución de VMWare Server no resultó tan estable como se esperaba. En una instalación realizada con tres máquinas virtuales con Linux y una con Windows, la plataforma sufría caídas recurrentes (llegando a caer prácticamente todos los días) y dejando inaccesibles las máquinas virtuales instaladas.
- Ventajas: Según la opinión de muchos administradores. Solidez, estabilidad, seguridad y soporte del fabricante ejemplar. En nuestras pruebas resulto ser bastante inestable.
- Desventajas: Dificultad de puesta en marcha para usuarios con escasas nociones, el gestor de máquinas virtuales tiene un rendimiento mediocre en máquinas con escaso hardware. Su código es propietario.
- Coste: Variable en función del producto. Existe una versión gratuíta.

VirtualBox

Sitio web: http://www.virtualbox.org/

- VirtualBox está disponible para Windows, OS X, Linux y Solaris.
- Luego de instalar vbox-additions, la integración entre el host (el sistema operativo del equipo físico) y el guest (el sistema operativo de la máquina virtual) es muy buena. Ofrece facilidades como portapapeles compartido, carpetas compartidas, modo fluido y redimensionamiento automático de la resolución/tamaño de ventana.
- Ventajas: Fácil administración de las máquinas. Se dispone del código bajo licencia GPL v2.
- Desventajas: No es posible modificar las propiedades de las máquinas virtuales mientras están en ejecución (memoria, tarjetas de red, discos, etc.). La administración de las

máquinas virtuales se debe realizar mediante un programa cliente instalado en el host. Bajo rendimiento.

- Coste: Gratuito

XEN

Sitio web: http://www.xen.org/

- Soporta modos de full y para-virtualization .
- Requiere que el hardware soporte virtualization technology (en caso de utilizar full virtualization) .
- La interfaz gráfica y la integración de ingreso y salida de datos es bastante mala. Utiliza una variación de VNC para el control de consola.
- Para máquinas virtuales Linux requiere que éstas utilicen un núcleo especializado, kernel-xen. Este kernel se puede instalar de manera nativa en distribuciones Red Hat (RHEL, CentOS y Fedora).
- El rendimiento con para-virtualization es bastante bueno en términos de uso de memoria, disco y CPU.
- El uso de discos raw (acceso directo a particiones o discos) es nativo. Esto elimina una capa adicional de acceso, utilizada comúnmente para gestionar archivos como discos virtuales.
- Una característica particular de Xen es que, al utilizar para-virtualization, el consumo de memoria RAM disminuye en el sistema operativo host al ser asignada a una máquina virtual.
- La configuración se realiza mediante un programa cliente instalado en el host, pero puede conectarse a la máquina virtual desde un cliente remoto.
- Es posible modificar el tamaño de memoria RAM asignada, conectar tarjetas de red y agregar discos en caliente.
- Orientado a usuarios más experimentados. Está desarrollado por la Universidad de Cambridge y por unidades de Intel y AMD. Cada vez más presente en diferentes distribuciones.
- Ventajas: Potente y escalable. Muy seguro. Sistema de para-virtualization innovador y efectivo. Desarrollo muy profesional.
- Desventajas: Curva de aprendizaje costosa, documentación no excesivamente abundante, tiempos de implantación mayores. No admite drivers de los entornos a emular. Desarrollo algo inmaduro.
- Coste: Gratuíto, es GPL.

QEMU/KVM

Sitio web: http://www.qemu.org / http://http://www.linux-kvm.org

Bastante conocido sobre todo entre los usuarios de soluciones Linux.

- Algunas aplicaciones pueden correr a una velocidad cercana al tiempo real.
- Soporte para ejecutar binarios de Linux en otras arquitecturas.
- Mejoras en el rendimiento cuando se usa el módulo del kernel KQEMU.
- Las utilidades de línea de comandos permiten un control total de QEMU sin tener que ejecutar X11.
- Control remoto de la máquina emulada a través del servidor VNC integrado.
- Soporte incompleto para Microsoft Windows como huésped y otros sistemas operativos (la emulación de estos sistemas es simplemente buena).

- Soporte incompleto de controladores(tarjetas de vídeo, sonido, E/S) para los huéspedes, por lo tanto se tiene una sobrecarga importante en aplicaciones multimedia.
- **Kernel-based Virtual Machine** o **KVM**, es una solución para implementar virtualización completa con Linux.
- KVM necesita un procesador x86 con soporte Virtualization Technology. Puede ejecutar huéspedes Linux (32 y 64 bits) y Windows (32 bits).
- Ventajas: Código libre, ligero en ejecución. Fácil de desplegar y configurar.
- Desventajas: Soporte escaso, desarrollo irregular, velocidad de CPU muy baja en entornos emulados. El consumo de recursos es mejorable.
- Coste: Gratuito, es GPL.

Característica \ Software	**VMWare**	**VirtualBox**	**Xen**	**Qemu/KVM**
Conocimiento requerido para administración	Medio	Bajo	Alto	Alto
Integración video, I/O	Medio	Alto	Bajo	Bajo
Capacidad de para-virtualización	No	No	Si	Parcial[1]
Driver para los guest	Si, vmware-tools	Si vbox-additions	No	No
Requerimientos del guest	Ninguno	Ninguno	Kernel-xen en para-virtualización	Ninguno
Discos Raw	Configuración adicional	Configuración adicional	Nativo	Nativo
Soporte Network Bridge	Si	Si	Si	Si
Sistemas Operativos guest probados	Windows XP, 2003 server Linux Ubuntu	Windows XP, 2003 Server, Linux Ubuntu	Windows XP, 2003 Server, Linux Ubuntu	Windows XP, 2003 server, 2008, Linux Ubuntu
Requiere configuración al hacer upgrade de Kernel	Si	Si	No	No
Migración en caliente	Si	No	Si	Si
Código	Propio	Disponible bajo GPL v2	Disponible bajo GPL v2	Disponible bajo GPL y LGPL
Coste	Variable, aunque existe una versión gratuita.	Gratuito	Gratuito	Gratuito

Tabla 7: Comparativa herramientas virtualización

[1] KVM no soporta para-virtualización para CPU pero puede soportar para-virtualización para otros dispositivos del sistema mejorando el rendimiento del sistema.

Resultados y discusión

Para realizar la comparativa práctica de las herramientas de virtualización seleccionadas hemos realizado varias mediciones que determinaran el rendimiento de cada una de ellas en un entorno controlado.

En pimer lugar hemos medido el consumo de CPU de un sistema Windows 2003 Server durante su arranque y el tiempo empleado en el mismo obteniendo los resultados mostrados en las tablas 8 y 9. Con ello podemos hacernos una idea del rendimiento de cada una de estas herramientas en uno de los momentos que este sistema operativo hace un uso intensivo del procesador.

	user	system	nice	Idle
KVM	51.20	21.9	0.00	11.1
Xen	2.23	3.89	0.00	52.65
VMware	11.29	59.2	0.00	11.4
VirtualBox	3.26	31.00	22.61	15.03

Tabla 8: Consumo de CPU en el arranque. Windows 2003 Server

	Tiempo (s)	CPU %
KVM	105	75.42
Xen	96	31.14
Vmware	92	72.9
VirtualBox	81	71.72

Tabla 9: Tiempo transcurrido durante el arranque. Windows 2003 Server

Otro de los parámetro a estudiar va a ser la cantidad de memoria que consume cada herramienta por tener estos sistemas ejecutándose, adicionalmente a la propia memoria RAM asignada al sistema cliente, tal como se puede ver en la tabla 10.

	MB
KVM	1294
Xen	1217
VMware	1331
VirtualBox	1198

Tabla 10: Consumo de RAM con la Máquina virtual en ejecución. Windows 2003 Server (1GB RAM)

En último lugar hacemos un estudio del rendimiento de cada una de estas herramientas sobre el sistema de ficheros virtualizado, comprobando los tiempos medios de acceso a ellos, tal como mostramos en las tablas 11 y 12.

	Tiempo	MB/s
KVM	1m03s	29,26
Xen	0m52s	35,45
VMware	0m59s	31,24
VirtualBox	1m12s	25,6

Tabla 11: Transferencia Disco. Tamaño fichero 1,8 GB

	Tiempo	MB/s
KVM	3m22s	9,12
Xen	2m48s	10,97
VMware	3m23s	9,08
VirtualBox	3m54s	7,88

Tabla 12: Tiempo en comprimir un fichero. Tamaño fichero 1,8 GB

Conclusiones

Las diferentes soluciones presentadas pueden ser utilizadas de manera óptima en diferentes entornos. En nuestro caso, la opción más recomendada es la plataforma Xen por los siguientes motivos:

- Mejor rendimiento y mejor soporte para servidores y host Linux.
- Escalabilidad y estabilidad.
- Permite tener máquinas virtuales con pocos recursos asignados
- Si bien la interfaz gráfica y la consola son bastante limitadas, esto no constituye una limitación, ya que la mayor parte de la interacción con el servidor puede realizarse mediante conexión remota vía SSH y/o RDP.

En otras circunstancias es posible que otra solución sea la más recomendada, sobre todo si no atendemos a la variable del coste:

- En este caso, la solución puede venir de la mano de VMWare, que ofrece una mayor cantidad de servicios de valor agregado como monitoreo, soporte en línea, consultoría y soporte local.

No obstante, somos conscientes de la gran cantidad de test de rendimiento que se pueden realizar sobre cada una de estas plataformas de virtualización, pero nos hemos limitado a los test indicados por diferentes motivos:

- **Discos:** Como hemos indicado en varias ocasiones a lo largo de este documento, el principal cuello de botella de estos sistemas es el disco duro, por lo tanto hemos realizado unos pequeños test para comprobar como se gestionan en cada caso.
- **Memoria:** La cantidad de memoria usada por cada plataforma, vendrá determinada por la memoria asignada a cada una de las máquinas en cuestión. Básicamente, ninguna de las plataformas elegidas incrementará este valor notablemente.
- **Uso CPU:** La CPU que podrá usarse por cada máquina esta limitada por la asignada en la configuración de la misma. En cuanto a la gestión de su uso, en sistemas para-virtualizados siempre obtendremos unos mejores valores, aunque para nuestro caso, no se trata de un parámetro determinante.
- Nuestro propuesta se basa en la optimización de recursos y la posibilidad de proveer a nuestros servidores de alta disponibilidad a bajo coste. En este proceso vamos a sacrificar gran parte del rendimiento de los sistemas, motivado principalmente por el uso de la virtualización. Por lo tanto nos interesan más las caracteristicas y carencias de las plataformas estudiadas antes que las pequeñas diferencias de rendimiento que puedan aportar.

Capitulo 5

Conclusiones y trabajo futuro

Conclusiones

Como primera y principal conclusión de esta tesis de Master podemos destacar la propuesta, implementación y expermimentación de un sistema de alta disponibilidad a bajo coste. Basándonos en la reutilización del hardware del que ya disponemos, todo ello mediante el uso de los sistemas de almacenamiento y plataformas de virtualización estudiados y con los que hemos experimentando durante la realización de este trabajo y se ha comprobado que aportan a la implementación final una estabilidad, fiabilidad y rendimiento más que aceptables para estos entornos y que detallaremos más adelante.

Todo ello nos va a permitir ofrecer a pequeñas, medianas y grandes empresas, que no puedan o quieran asignar excesivos recursos económicos a este fin, poder disponer de una serie de servicios/servidores disponibles sin depeder completamente del hardware en el que esten ejecutandose y permitiendo la restauración de los mismos en otros equipos en caso de desastre inminente.

En un primer lugar, hemos estudiado la posibilidad de disponer de un sistema de almacenamiento distribuido y accesible desde diferentes sistemas físicos así como contrastado diferentes sistemas de ficheros que nos dan la posibilidad de acceso concurrente a estos recursos. Hemos podido apreciar tras comparar costes y eficiencia que lo que más se adapta a lo que inicialmente estamos buscando es una combinación de hardware/software basado en la utilización de discos locales (los mismo que ya se suelen tener en los servidores de la empresa) y la tecnología/software DRBD que nos permitirá crear un raid 1 (mirroring) a través de la red, proporcionándonos seguridad y a la vez combinado con un sistema de archivos de cluster tipo ocfs2 y/o gfs2, la posibilidad de acceder a los mismos desde dos nodos diferentes de nuestro cluster.

En segundo lugar, hemos estudiado diferentes sistemas de virtualización para ejecutar en ellos los servidores/servicios que queramos ofrecer y ver cual se adapta mejor a nuestro proposito. En este punto podemos destacar como platarforma recomendada Xen, aunque en algunos casos, por otros motivos como la elección del sistema host o por el sistema de almacenamiento puede llegar a ser una mejor opción optar por plataformas como Qemu/KVM.

Por último, hemos estudiado diferentes sistemas de clustering que ofrezcan alta disponibilidad, descartando directamente los que necesitan una inversión elevada para llegar a obtener las prestaciones requerídas, como son las soluciones VMWare VSphere y XenServer. De las dos estudiadas podemos decir que son perfectamente válidas y que la elección entre una u otra puede venir dada más por elecciones externas como el sistema de ficheros de cluster o la predilección por una u otra herramienta de configuración.

Trabajo futuro

Actualmente, cada uno de los puntos tratados en esta tesis de Master está siendo aplicado por gran cantidad de grupos de trabajo. Estos trabajos, nos ofrecen una gran variedad de herramientas y plataformas para construir clusters y poder ofrecer gran cantidad de servicios los cuales dispongan de alta disponibilidad.

Posibles ampliaciones de esta tesis de Master podrían ser el estudio de muchos más sistemas de archivos de cluster disponibles y sistemas de almacenamiento como nutanix [EMP002], al igual que el sistema de virtualización opensource como openVZ [VIRT009] que no se han realizado por no disponer de dicho hardware y /o suponer un coste económico no asumible.

Dado que el punto más crítico a la hora de ofrecer alta disponibilidad de servidores virtualizados es en todo momento el almacenamiento, sería interesante estudiar más a fondo cada uno de los sitemas de almacenamiento y ficheros propuestos para buscar alternativas que ofrecieses opciones como snapshots y copias de seguridad de los discos de los sistemas virtualizados, así como estudiar la posibilidad de llegar a minimizar las ventanas temporales en los que es preciso apagar o reiniciar los sitemas virtualizados.

Tal como comentabamos en la propuesta de este trabajo, otro campo abierto, es la posibilidad de crear clusters de servidores virtualizados, estudiando las multiples ventajas que nos pueden aportar las plataformas de virtualización para dotar a clusters de alta disponibilidad de una granja de servidores virtualizados mediante los cuales mantener nuestros servicios siempre activos.

Bibliografía

[REDIRIS001] "Alta disponibilidad gracias a las tecnologías de virtualización y redes"
Josep Vidal Canet, Sergio Cubero Torres
Servicio de Informática - Universidad de Valencia
http://www.rediris.es/jt/jt2007/presentaciones/P14B.ppt

[HA001] Pacemaker: http://clusterlabs.org/wiki/Documentation

[HA002] RedHat Cluster Suite: http://www.redhat.com/docs/manuals/csgfs/

[HA003] RedHat Cluster Suite:
http://www.centos.org/docs/5/html/5.1/Cluster_Suite_Overview/index.html

[HA004] Linux-HA: http://www.linux-ha.org/

[HA005] Service Availability standards: http://www.saforum.org/

[HA006] OpenAIS: http://freecode.com/projects/openais

[HA007] Corosync: http://www.corosync.org/doku.php?id=welcome

[HA008] Paulo Clavijo Esteban
Consultor de sistemas Linux y tecnologías J2EE.
http://www.lintips.com/

[VIRT001] Xen: http://www.xen.org/support/documentation.html

[VIRT002] KVM: http://www.linux-kvm.org/page/Documents

[VIRT003] QEMU: http://wiki.qemu.org

[VIRT004] VMWare: http://www.vmware.com/es/virtualization/
"Virtualización y gestión de infraestructuras"

[VIRT005] Hyper-V: http://www.microsoft.com/es-xl/servidores-nube/windows-server/default.aspx
"Windows Server 2008 R2 Hyper-V"

[VIRT006] MED-V: http://www.microsoft.com/es-es/windows/enterprise/products-and-technologies/mdop/med-v.aspx
Microsoft Enterprise Desktop Virtualization (MED-V)

[VIRT007] VirtualBox: https://www.virtualbox.org/

[VIRT008] Parallels Desktop: http://www.parallels.com/es/products/desktop/

[VIRT009] OpenVZ: http://wiki.openvz.org/Main_Page

[VIRT010] Microsoft Virtual PC: http://www.microsoft.com/windows/virtual-pc/default.aspx

[VIRT011] http://www.intel.com/technology/itj/2006/v10i3/1-hardware/6-vt-x-vt-i-solutions.htm
Intel VT

[VIRT012] http://sites.amd.com/es/business/it-solutions/virtualization/Pages/amd-v.aspx
AMD-V

[VIRT013] http://www.citrix.es/Productos_y_Soluciones/Productos/XenApp/
XenApp

[VIRT014] "VDI Smackdown!"
Ruben Spruijt
http://www.pqr.com

[VIRT015] "Consumerización de las TI"
InfoWorld. Custom Solutions Group.
http://www.citrix.com/byo

[VIRT016] http://en.wikipedia.org/wiki/Logical_partition_(virtual_computing_platform)
Logical Partition, LPAR

[EMP001] http://www-05.ibm.com/services/es/bcrs/a1026936.html
Servicios de Alta Disponibilidad
Servicios TI – IBM. 2012

[EMP002] http://www.nutanix.com/
Sistemas de almacenamiento

[WIKI001] http://es.wikipedia.org/wiki/Cl%C3%BAster_de_alta_disponibilidad
Clusters de Alta Disponibilidad.

[WIKI002] http://en.wikipedia.org/wiki/IBM_M44/44X
IBM M44/44X

[WIKI003] http://en.wikipedia.org/wiki/OS/VS2
OS/VS1, OS/VS2, MVS

[WIKI004] http://en.wikipedia.org/wiki/Computer_cluster
Tipos de Cluster

[SF001] http://wiki.lustre.org/index.php/Preparing_to_Install_Lustre
Lustre

[DRBD001] http://www.drbd.org/
DRBD

[BENCH001] http://www.iozone.org/
Iozone. Herramienta de benchmark (sistemas de almacenamiento)

[BENCH002] http://www.coker.com.au/bonnie++/
Bonnie++. Herramienta de benchmark (sistemas de almacenamiento)

[HD001] http://es.wikipedia.org/wiki/Serial_Attached_SCSI
Serial Attached SCSI, SAS

[HD002] http://es.wikipedia.org/wiki/Serial_ATA
SATA

[HD003] http://es.wikipedia.org/wiki/Ssd
Unidades de Estado Solido

Anexo I. Herramientas de análisis de prestaciones en discos.

Existen multitud de herramientas para evaluar las prestaciones de los diferentes tipos de almacenamiento y sus sistemas de ficheros. En esta tesina nos vamos a basar principalmente en herramientas bajo Linux, ya que es el sistema sobre el que trabajamos para conseguir un sistema de bajo costo.

Con estas aplicaciones vamos a poder medir el rendimiento o alcance de un determinado sistema de ficheros (ext4, ocfs2, gfs, etc) o una clase concreta de acceso a datos (NFS, iSCSI, DRBD, etc), obteniendo información como la velocidad de lectura/escritura, latencias, etc.

cp

Comando utilizado para copiar archivos desde un origen a un destino. La siguiente orden medirá el tiempo empleado en escribir un determinado fichero de X * Y bytes en disco.

```
# dd if=/dev/zero of=file bs=X count=Y && time cp file /mnt/tmp
```

hdparm

hdparm es una pequeña herramienta que sirve para manipular la configuración de las unidades de disco además de medir el rendimiento de las mismas. El objetivo es optimizar el tiempo de acceso o la velocidad de transferencia.

Sintaxis: hdparm -tT /dev/[disco]

dd

Permite la copia de datos bit a bit independientemente del sistema de ficheros origen y destino. Como información de salida ofrece la velocidad de lectura/escritura con la que se ha realizado la operación, así como el tiempo empleado.

El formato que utilizaremos para el comando dd será el siguiente:

- Lectura de X * Y bytes desde el disco */dev/sdx* al dispositivo */dev/null*.

```
# dd if=/dev/sdx of=/dev/null bs=X count=Y
```

- Escritura de X * Y bytes desde el dispositivo origen */dev/zero* al fichero */tmp/output*.

```
# dd if=/dev/zero of=/tmp/output bs=X count=Y
```

El dispositivo */dev/null* descarta toda la información volcada en él de forma instantánea. De esta forma, al aplicar el primer comando el tiempo de cómputo se invertirá exclusivamente en la lectura de datos.

El dispositivo */dev/zero* es un dispositivo especial que cuando se lee de él proporciona caracteres NULL, es decir, su tiempo de lectura es prácticamente nulo. De esta forma, al aplicar el segundo comando el tiempo de cómputo se invertirá exclusivamente en la escritura de datos.

Una forma habitual de uso de dd es por ejemplo leer y escribir un fichero superior a la memoria RAM total (por ejemplo 5 GB en caso de disponer de 4 GB de memoria RAM). Realizando esta operación varias veces, tanto en lectura (R) como en escritura (W) variando los valores de X e Y de tal forma que su multiplicación dé siempre el mismo tamaño de archivo podremos obtener una

media que se aproxime más a la realidad.

iozone

IOzone [BENCH001] es una herramienta de benchmark destinada a comprobar el rendimiento de un sistema de archivos. La aplicación genera y mide una gran cantidad de operaciones sobre ficheros.

A continuación vamos a exponer las opciones más usadas de este comando de cara a medir el rendimiento de un sistema de archivos:

- *-a*: modo automático completo.
- *-b excel_file*: genera un archivo en formato Excel con los resultados obtenidos (formato binario).
- *-c*: incluye en los resultados los tiempos empleados para cerrar los ficheros.
- *-f filename*: especifica el archivo temporal que se empleará para los tests.
- *-g size*: tamaño máximo del archivo (Ej. -g 2G, igual o superior al tamaño de la memoria RAM) para el modo auto (*-a*).
- *-i test*: especifica el tipo de test a utilizar (*0*: escritura/re-escritura, *1*: lectura/re-lectura, *2*: lectura/escritura aleatoria, ...).
- *-r size*: tamaño fijo del registro utilizado para hacer las transferencias.
- *-s size*: tamaño fijo del archivo utilizado para hacer las transferencias.
- *-z*: esta opción utilizada junto con el parámetro -a, fuerza a IOzone a emplear archivos pequeños durante las pruebas.
- *-R*: genera un informe Excel.

Las distintas definiciones de los tests son las siguientes:

- Escritura: mide el rendimiento de escritura secuencial en un nuevo archivo.
- Re-escritura: mide el rendimiento de escritura secuencial sobre un archivo que ya existe.
- Lectura: mide el rendimiento de lectura secuencial sobre un archivo existente.
- Re-lectura: mide el rendimiento de lectura secuencial sobre un archivo que ha sido recientemente leído.
- Escritura aleatoria: mide el rendimiento de escritura aleatoria en un nuevo archivo.
- Lectura aleatoria: mide el rendimiento de lectura aleatoria sobre un archivo ya existente.
- Fescritura: mide el rendimiento de escritura usando la función de librería fwrite()
- Fre-escritura: mide el rendimiento de escritura usando la función de librería fwrite() sobre un archivo que ya existe.
- Flectura: mide el rendimiento de lectura usando la función de librería fread()
- Fre-lectura: mide el rendimiento de lectura usando la función de librería fread() sobre un archivo que ya existe.

Una forma habitual de uso de IOzone puede ser la siguiente:

```
# iozone -Razc -i 0 -i 1 -g 5G -b fichero.xls
```

La orden anterior ejecuta un test de lectura/escritura secuencial sobre el dispositivo que corresponda al directorio en el que estemos situados, variando los tamaños de los ficheros empleados desde 64 KB a 5 GB, utilizando buffers de transferencia comprendidos entre 4 KB y 16 MB. Como resultado final se generará un fichero Excel a partir del cual podremos crear las

correspondientes gráficas.

Esto último es lo que se conoce como barrido de un determinado espectro de almacenamiento, y tiene como principal objetivo el de poder generar una gráfica de superficie que determine el comportamiento del sistema ante variaciones de los registros y ficheros empleados.

En esta clase de pruebas es conveniente utilizar un tamaño de fichero máximo (parámetro *-g*) superior a la memoria RAM disponible. De esta forma se podrá obtener el comportamiento de la máquina cuando utiliza la memoria caché del procesador (tamaño del archivo inferior a la caché del procesador), cuando emplea la memoria RAM (tamaño del archivo comprendido entre la caché del procesador y la cantidad total de memoria RAM) o cuando hace uso directamente de las operaciones de entrada/salida a disco (tamaño del archivo superior a la memoria RAM).

Otra forma habitual de empleo de IOzone está destinada a medir el impacto de diferente número de procesos sobre dicho sistema de archivos. Por ejemplo, la orden siguiente hará que varios procesos (desde 1 a 50) vayan realizando operaciones de lectura/escritura secuencial de un fichero de 4 MB, utilizando para ello registros de 64 KB.

```
# iozone -Rc -r 64 -s 4MB -l 1 -u 50 -i 0 -i 1 -b /root/fichero.xls
```

El ejemplo anterior se suele utilizar para obtener el comportamiento de múltiples procesos del sistema manejando archivos pequeños. Para completar las pruebas de IOzone es conveniente repetir este test pero utilizando archivos más grandes (y aumentando a su vez el tamaño de los registros).

```
# iozone -Rc -r 1024 -s 512MB -l 1 -u 12 -i 0 -i 1 -b /root/fichero.xls
```

Bonnie++

Bonnie++ [BENCH002] permite la creación de distintos tests de lectura, escritura y borrado de archivos de diversos tamaños, etc. Bonnie++ tiene muchos parámetros que pueden ser empleados, pero la forma en la que lo utilizaremos para medir el rendimiento de un sistema de ficheros será la siguiente:

```
# bonnie++ -d /tmp [-n number:max:min:num-directories] [-s size] -u 0
```

A través de la opción *-d* se especifica el directorio a utilizar durante el transcurso de las pruebas. El parámetro *number* es el número de archivos que serán creados multiplicado por 1024, el cual vendrá precedido de la opción -n. Si se especifican los valores *max* y *min*, los archivos serán creados con un tamaño aleatorio comprendido entre esas dos cotas (bytes). Y si de declara el parámetro *directories*, los archivos serán distribuidos uniformemente a través de una estructura de directorios con una profundidad máxima marcada por el parámetro *directories*.

Con la opción *-s* se indica a través del valor *size*, el tamaño del archivo que será utilizado para las pruebas de lectura y escritura. Este tamaño debe ser como mínimo el doble de la memoria RAM. Y con la opción *-u 0* se indica a Bonnie++ que ejecute las pruebas como usuario root.

Los resultados mostrados por Bonnie++ ofrecen velocidades de escritura secuencial (*Sequential Output*) y lectura secuencial (*Sequential Input*). También ofrecen valores sobre la creación secuencial (*Sequential Create*) y aleatoria (*Random Create*) de ficheros, así como de borrado secuencial (*Sequential Delete*) y aleatorio (*Random Delete*) de ficheros.

Bonnie++ trae consigo una herramienta que nos permite generar un fichero en formato HTML

(*bon_csv2html*) a partir de los resultados obtenidos. Para ello deberemos ejecutar la siguiente secuencia de órdenes:

```
# echo "última_línea_de_datos"  bon_csv2html > fichero.html
```

Una forma habitual de uso de Bonnie++ puede ser la siguiente:

```
# bonnie++ -n 128 -s 7544 -x 1 -u 0 -d /mnt/tmp/
```

La orden anterior ejecuta una sola vez (*-x 1*), un test de lectura y escritura de un archivo de 7,5 GB, así como la tarea de creación y borrado de 131072 ficheros (128x1024) de 0 bytes.

Anexo II. Resultados tests sistemas de almacenamiento.

En el apartado *Resultados y Discusiones* del *Estudio de prestaciones en sistemas de almacenamiento en clusters,* se han introducido las gráficas más representativas para la comparativa que hemos realizando. Pero debido a la cantidad de datos y gráficas obtenidas durante la pruebas de rendimiento de los sistemas de almacenamiento y sistemas de ficheros, que consideramos de gran interés para un análisis de los mismos, en este anexo vamos a detallar los archivos y su contenido que se han guardado en el CD adjunto.

Sistemas sobre los que se han realizado las pruebas.

Sistema 1:

Quad core
4GB RAM
Disco Sistema 500GB 7.2k. sataII
Disco Datos 500GB 7.2k. sataII
2x Tarjeta Intel 1Gb/s

Sistema 2:

2 x Quad core
16GB RAM
Disco Sistema 2 x 500GB en RAID1 HW. 7.2k sataII
Discos Datos 4 x 500GB en RAID5 HW. 7.2k sataII
Discos Datos 4 x 500GB en RAID5 HW. 15K7 sas
Controladora RAID SAS/SATA LSI SAS2108
Tarjeta Intel Ethernet Server Adapter X520-LR1 E10G41BFLR Fibra Monomodo 10Gb/s

NAS

NAS NSS4000 4-Bay Gigabit de Cisco con 4 discos de 500GB 7.2k en RAID5.

Los archivos .wks contienen los datos obtenidos y se trata de los archivos generados por la herramienta iozone.

Los archivos .xls contienen las gráficas generadas a partir de los datos anteriores.

lab_ext4.wks, lab_ext4_graphs.xls : Pruebas de rendimiento realizadas sobre la partición de datos del sistema 1, con sistema de ficheros ext4. Esta prueba la hemos realizado para tener un punto de referencia frente a los sistemas de ficheros de cluster que vamos a usar.

lab_nfs_ext4.wks, lab_nfs_ext4_graphs.xls : Pruebas de rendimiento realizadas sobre una unidad nfs del NAS montada sobre el sistema 1, con sistema de ficheros ext4. Prueba realizada para comparar el rendimiento de un sistema nfs frente a un sistema de archivos local.

lab_drbd_gfs2.wks, lab_drbd_gfs2_graphs.xls : Pruebas de rendimiento realizadas sobre la partición de datos del sistema 1, con drbd y sistema de ficheros gfs2. Prueba realizada para comparar el rendimiento entre los dos sistema de ficheros de cluster estudiados (ocfs2 y gfs2).

lab_drbd_ocfs2.wks, lab_drbd_ocfs2_graphs.xls : Pruebas de rendimiento realizadas sobre la partición de datos del sistema 1, con drbd y sistema de ficheros ocfs2. Prueba realizada para comparar el rendimiento entre los dos sistema de ficheros de cluster estudiados (ocfs2 y gfs2).

lab_drbd_ocfs2_sataii.wks, lab_drbd_ocfs2_sataii_graphs.xls : Pruebas de rendimiento realizadas sobre la partición de datos en los discos sataII del sistema 2, con drbd y sistema de ficheros ocfs2. Prueba realizada para comparar entre el rendimiento de los discos sas y sataII bajo las mismas condiciones.

lab_drbd_ocfs2_sas.wks, lab_drbd_ocfs2_sas_graphs.xls : Pruebas de rendimiento realizadas sobre la partición de datos en los discos sas del sistema 2, con drbd y sistema de ficheros ocfs2. Prueba realizada para comparar entre el rendimiento de los discos sas y sataII bajo las mismas condiciones.