UNIVERSITAT POLITÈCNICA DE VALÈNCIA

UNIVERSIDAD POLITÉCNICA DE VALENCIA

Máster en Ingeniería de Computadores

INSTALACIÓN Y CONFIGURACIÓN DE UN CLUSTER DE ALTA DISPONIBILIDAD CON REPARTO DE CARGA

SERVIDOR WEB Y MAQUINAS VIRTUALES

Alumno: Lenin Alcántara Roa.

Director: Pedro López Rodríguez.

Febrero 2014

ÍNDICE

1. INTRODUCCION

Los entornos de computación actuales necesitan varios ordenadores para resolver las tareas que una sola no sería capaz de manejar. Los grandes entornos de computación actuales implican el uso de grandes grupos de servidores donde cada nodo está conectado con el resto en un entorno de Clustering.

La computación en Clustering, en su nivel más básico, involucra dos o más ordenadores como servidores para un único recurso. Las aplicaciones están convirtiendo al Clustering como una vía para aumentar la carga de datos. La práctica de distribuir atributos desde una simple aplicación en varias computadoras no solo aumenta la eficiencia, sino también la redundancia en caso de fallo. Un primer ejemplo de un Cluster básico es el DNS (Domain Name Service, Servicio de nombres de dominio), que se construye en las caches primarias y secundarias de los servidores. Otros protocolos también se han construido con características de Clustering-Redundante, como NIS y SMTE.

Aunque para muchos el Clustering no sea la panacea para los problemas actuales, este mecanismo puede ayudar a una organización que está intentando maximizar algunos de sus recursos existentes. Aunque no todos los programas se puedan beneficiar del Clustering, las organizaciones que proporcionan aplicaciones como servidores web, bases de datos y servidores FTP, sí se pueden beneficiar de la tecnología según aumenta la carga del sistema. Los Clusters se pueden diseñar fácilmente teniendo en mente la escalabilidad; según aumentan los requisitos se pueden añadir más sistemas, lo que nos permite repartir la carga por múltiples máquinas.

Debido a todas las ventajas y beneficios que proporcionan las técnicas de Clustering en el entorno tecnológico actual, este proyecto que presentamos se fundamenta en la implementación y configuración de un Cluster para ofrecer servicio web y máquinas virtuales. Adicionalmente, y como una medida de tolerancia a fallos se ha incorporado al Cluster un sistema de Equilibrado de Carga con Alta Disponibilidad (HA), algo que resulta beneficioso en el sentido de que todos los nodos del sistema aportan su capacidad tecnología de manera equitativa para ofrecer los servicios de forma más veloz.

Para la implementación y configuración del Cluster hemos construido un conjunto de herramientas automatizadas, que permiten la instalación de manera desatendida de todos los nodos que componen el Cluster. Esto es bastante factible, debido a que en entornos de Clustering donde existe una gran cantidad de nodos nos ayuda a realizar las tareas de mantenimiento de una forma más rápida y sofisticada.

1.1 Objetivos.

El objetivo principal de este proyecto consiste en la instalación y configuración de un Cluster de computadores, el cual alojara un servidor web de Alta Disponibilidad y Reparto de Carga. También, se ha decido ofrecer a sus usuarios un servicio de máquinas virtuales.

Otro de nuestros objetivos planteados en este proyecto es la implementación de herramientas (scripts) que permitan la instalación, administración y mantenimiento de los sistemas del Cluster, de una forma automática y desatendida.

1.2 Motivación.

El vertiginoso crecimiento que ha tenido la tecnología de Clustering en los últimos años es muy considerable. En la actualidad esta tecnología se ha convertido en una alternativa bastante factible y beneficiosa en diferentes entornos y múltiples organizaciones. Tanto a nivel científico, educativo y empresarial la tecnología de Clustering tiene una gran aceptación y bastante interés de desarrollo.

El hecho de incorporar al Cluster un sistema de máquinas virtuales es debido al gran auge y repunte de la tecnología de virtualización en las últimas décadas. En la actualidad la virtualización de sistemas está jugando un papel protagónico en el ofrecimiento de servicios tecnológicos. Desde hace varios años organizaciones de todos los ámbitos están empleando sistemas de virtualización para optimizar sus servicios.

1.3 Resumen.

La técnica de Clustering es un mecanismo que permite el agrupamiento de un conjunto de ordenadores y dispositivos de entrada/salida para compartir recursos y trabajar como una sola máquina. Mayormente los nodos que componen un Cluster son servidores de alto rendimiento y con capacidad superior que un ordenador de sobremesa. Aunque, pueden existir Clusters con ordenadores de bajo rendimiento también.

En este proyecto se ha implementado un Cluster de Alta Disponibilidad con Reparto (Equilibrado) de la Carga. Este tipo de Clustering es bastante factible en entornos críticos donde la provisión del servicio que ofrece la máquina debe estar presente en todo momento. En este Cluster se ha implementado un servidor Web y un Sistema de Máquinas Virtuales para los usuarios del mismo.

2. ESTADO DEL ARTE

En este capítulo se describe el estado tecnológico que rodea a los *Clusters* en la actualidad. Realizamos una evaluación de varios Sistemas Operativos para servidores de acuerdo a las características que ofrecen cada uno, valorando las ventajas y desventajas que proporcionan.

En este mismo orden, detallamos las razones por las que hemos utilizado un Sistema Operativo de Código Abierto (Open Source) para realizar nuestro proyecto, y los beneficios que proporciona tener un Cluster de Alta Disponibilidad.

También, ponemos de manifiesto los beneficios que aporta a una empresa o institución el uso de Clusters como herramienta clave en su infraestructura tecnológica.

2.1 ¿Qué es un Cluster?

El término Cluster se aplica a los conjuntos o conglomerados de computadoras construidos mediante la utilización de hardware común y que se comportan como si fuesen una única computadora.

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

El cómputo con Cluster surge como resultado de la convergencia de varias tendencias actuales que incluyen la disponibilidad de microprocesadores económicos de alto rendimiento y redes de alta velocidad, el desarrollo de herramientas de software para cómputo distribuido de alto rendimiento, así como la creciente necesidad de potencia computacional para aplicaciones que la requieran.

Simplemente, un 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 que los comunes de escritorio.

Los Cluster son usualmente empleados para mejorar el rendimiento y/o la disponibilidad por encima de la que es provista por un solo computador típicamente siendo más económico que computadores individuales de rapidez y disponibilidad comparables.

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

- Alto rendimiento
- Alta disponibilidad
- Balanceo de carga
- Escalabilidad

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

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

- **Beneficios de la tecnología Cluster.**

Las aplicaciones paralelas escalables requieren: buen rendimiento, baja latencia, comunicaciones que dispongan de gran ancho de banda, redes escalables y acceso rápido a archivos. Un Cluster puede satisfacer estos requisitos usando los recursos que tiene asociados a él.

Los Clusters ofrecen las siguientes características a un costo relativamente bajo:

1. Alto rendimiento
2. Alta disponibilidad
3. Alta eficiencia
4. Escalabilidad

La tecnología Cluster permite a las organizaciones incrementar su capacidad de procesamiento usando tecnología estándar, tanto en componentes de hardware como de software que pueden adquirirse a un costo relativamente bajo.

- **Clasificación de los Clusters.**

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

- HPCC (High Performance Computing Clusters: Clusters de Alto Rendimiento).
- HA o HACC (High Availability Computing Clusters: Clusters de Alta Disponibilidad).
- HT o HTCC (High Throughput Computing Clusters: Clusters de Alta Eficiencia).

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

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

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

- **Los Clusters se pueden también clasificar en:**

Clusters de IT comerciales (de Alta Disponibilidad y Alta Eficiencia) y Clusters científicos (de Alto Rendimiento).

A pesar de las discrepancias a nivel de requisitos de las aplicaciones, muchas de las características de las arquitecturas de hardware y software, que están por debajo de las aplicaciones en todos estos Clusters, son las mismas. Más aún, un Cluster de determinado tipo, puede también presentar características de los otros.

- **Componentes de un Cluster.**

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

1. Nodos.
2. Almacenamiento.
3. Sistema Operativo.
4. Conexiones de red.
5. Middleware.
6. Elementos auxiliares: PDU, Conmutador KVM, Rack.
7. Ambientes de programación paralela.

1. **Nodos.**

Pueden ser simples ordenadores, sistemas multiprocesador o estaciones de trabajo (Workstations). Un nodo es un punto de intersección o unión de varios elementos que confluyen en el mismo lugar.

El Cluster puede estar conformado por nodos dedicados o por nodos no dedicados.

En un Cluster con nodos dedicados, los nodos no disponen de teclado, ratón ni monitor y su uso está exclusivamente dedicado a realizar tareas relacionadas con el Cluster. Mientras que, en un Cluster con nodos no dedicados, los nodos disponen de teclado, ratón y monitor y su uso no está exclusivamente dedicado a realizar tareas relacionadas con el Cluster, el Cluster hace uso de los ciclos de reloj que el usuario del computador no está utilizando para realizar sus tareas.

Cabe aclarar que a la hora de diseñar un Cluster, los nodos deben tener características similares, es decir, deben guardar cierta similaridad de arquitectura y sistemas operativos, ya que si se conforma un Cluster con nodos totalmente heterogéneos (existe una diferencia grande entre capacidad de procesadores, memoria, disco duro) será ineficiente debido a que el middleware delegará o asignará todos los procesos al nodo de mayor capacidad de cómputo y solo distribuirá cuando este se encuentre saturado de procesos; por eso es recomendable construir un grupo de ordenadores lo más similares posible.

2. Almacenamiento.

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

Tecnologías en el soporte del almacenamiento en discos duros:

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

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

Por su parte, DAS (Direct Attached Storage) consiste en conectar unidades externas de almacenamiento SCSI o a una SAN (Storage Area Network: “Red de Area de Almacenamiento”) a través de un canal de fibra. Estas conexiones son dedicadas.

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

3. Sistema operativo.

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

Ejemplos:

- **GNU/Linux**
  1. Ubuntu Server
  2. Fedora
  3. CentOS
  4. OpenMosix
  5. Sun Grid Engine

- **Unix**
  1. Solaris
  2. HP-UX
  3. AIX

- **Windows**
  1. NT
  2. 2000 Server
  3. 2003 Server
  4. 2008 Server

- **Mac OS X**

4. Conexiones de red.

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

- **Ethernet:**

Son las redes más utilizadas en la actualidad, debido a su relativo bajo coste. No obstante, su tecnología limita el tamaño de paquete, realizan excesivas comprobaciones de error y sus protocolos no son eficientes, y sus velocidades de transmisión pueden limitar el rendimiento

de los Clusters. Para aplicaciones con paralelismo de grano grueso puede suponer una solución acertada.

La opción más utilizada en la actualidad es Gigabit Ethernet (1 Gbit/s), siendo emergente la solución 10 Gigabit Ethernet (10 Gbit/s). La latencia de estas tecnologías está en torno a los 30 a 100 µs, dependiendo del protocolo de comunicación empleado.

En todo caso, es la red de administración por excelencia, así que aunque no sea la solución de red de altas prestaciones para las comunicaciones, es la red dedicada a las tareas administrativas.

- **Myrinet:**

Myrinet es una red de interconexión de clusters de altas prestaciones. Sus productos han sido desarrollados por **Myricom** desde 1995.

  - **Características**

Myrinet físicamente consiste en dos cables de fibra óptica, upstream y downstream, conectados con un único conector. La interconexión se suele realizar mediante conmutadores y encaminadores. Estos dispositivos suelen tener capacidades de tolerancia a fallos, con control de flujo, control de errores y monitorización de la red. Desde su creación se ha incrementado su rendimiento hasta alcanzar latencias de 3 microsegundos y anchos de banda de 10 Gbit/s:

- La primera generación de productos Myrinet obtenía anchos de banda de 512 Mbit/s
- La segunda de 1280 Mbit/s
- Myrinet 2000 obtiene 2 Gbit/s
- Myri-10G llega a los 10 Gbit/s, y puede interoperar con 10Gb Ethernet

Una de sus principales características, además de su rendimiento, es que el procesamiento de las comunicaciones de red se hace a través de chips integrados en las tarjetas de red de Myrinet (Lanai chips), descargando a la CPU de gran parte del procesamiento de las comunicaciones.

En cuanto al middleware de comunicación, la inmensa mayoría está desarrollada por Myricom.

  - **Usos**

Las especiales características de Myrinet hacen que sea altamente escalable, gracias a la tecnología existente de conmutadores y routers, y su presencia en el tramo de clusters de gran tamaño es importante. Muchos de los supercomputadores que aparecen en el TOP500 utilizan esta tecnología de comunicación.

Por ejemplo, los supercomputadores que forman parte de la Red Española de Supercomputación (dos de los cuales, Magerit y Marenostrum, están incluidos en el TOP500) utilizan Myrinet como red de interconexión para el paso de mensajes MPI.

- **InfiniBand:**

Es una red surgida de un estándar desarrollado específicamente para realizar la comunicación en Clusters.

Al igual que Fibre Channel, PCI Express y otros modos de interconexión modernos, InfiniBand usa un bus serie bidireccional de tal manera que evita los problemas típicos asociados a buses paralelos en largas distancias (en este contexto, una habitación o edificio). A pesar de ser una conexión serie, es muy rápido, ofreciendo una velocidad bruta de unos 2,5 Gigabits por segundo (Gbps) en cada dirección por enlace. InfiniBand también soporta doble e incluso cuádruples tasas de transferencia de datos, llegando a ofrecer 5 Gbps y 10 Gbps respectivamente. Se usa una codificación 8B/10B, con lo que, de cada 10 bits enviados solamente 8 son de datos, de tal manera que la tasa de transmisión útil es 4/5 de la media. Teniendo esto en cuenta, los anchos de banda ofrecidos por los modos simples, doble y cuádruple son de 2, 4 y 8 Gbps respectivamente.

Los enlaces pueden añadirse en grupos de 4 o 12, llamados 4X o 12X. Un enlace 12X a cuádruple ritmo tiene un caudal bruto de 120 Gbps, y 96 Gbps de caudal eficaz. Actualmente, la mayoría de los sistemas usan una configuración 4X con ritmo simple, aunque los primeros productos soportando doble ritmo ya están penetrando en el mercado. Los sistemas más grandes, con enlaces 12X se usan típicamente en lugares con gran exigencia de ancho de banda, como clústeres de computadores, interconexión en superordenadores y para interconexión de redes.

5. Middleware.

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

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

Existen diversos tipos de middleware, como por ejemplo: MOSIX, OpenMosix, Cóndor, OpenSSI, etc.

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

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

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

6. Elementos Auxiliares.

En una arquitectura de Clustering existen varios elementos auxiliares que componen la estructura de la máquina. Entre esto podemos destacar los siguientes:

- **PDU (Power Distribution Unit):**

Una unidad de distribución de energía (PDU) es un dispositivo equipado con salidas múltiples, diseñados para distribuir la energía eléctrica, especialmente a los racks de ordenadores y equipos de red ubicados dentro de un cluster. Las PDU varían de "regletas" simples y de bajo costo de montaje en rack, hasta PDU montadas en el suelo. Ofrecen múltiples funciones, incluyendo el filtrado de potencia para mejorar la calidad de la energía, el equilibrio de carga inteligente, monitoreo remoto y control mediante LAN o SNMP.

- **Conmutador KVM:**

Un switch KVM (Keyboard-Video-Mouse) es un dispositivo de computación que permite el control de distintos equipos informáticos con un solo monitor, un único teclado y un único ratón. Este dispositivo permite dotar al puesto de trabajo de tan sólo una consola para manejar al mismo tiempo varios PC o servidores, conmutando de uno a otro según sea necesario.

- **Rack:**

Un rack es un soporte metálico destinado a alojar equipamiento electrónico, informático y de comunicaciones. Las medidas para la anchura están normalizadas para que sean compatibles con equipamiento de cualquier fabricante. También son llamados bastidores, cabinas, cabinets o armarios.

Externamente, los racks para montaje de servidores tienen una anchura estándar de 600 mm y un fondo de 600, 800, 900, 1000 y ahora incluso 1200mm. La anchura de 600 mm para racks de servidores coincide con el tamaño estándar de las losetas en los centros de datos. De esta manera es muy sencillo hacer distribuciones de espacios en centros de datos (CPD). Para el cableado de datos se utilizan también racks de 800 mm de ancho, cuando es necesario disponer de suficiente espacio lateral para el guiado de cables.

7. **Ambientes de Programación Paralela.**

Los ambientes de programación paralela permiten implementar algoritmos que hagan uso de recursos compartidos: CPU (Central Processing Unit: "Unidad Central de Proceso"), memoria, datos y servicios.

2.2 Clustering de Alta Disponibilidad con Linux.

En los tiempos que vivimos es vital para las empresas y el negocio tener las aplicaciones, servicios y páginas web disponibles siempre, y no sólo hablamos por el tema económico, sino por cuestiones de imagen. Por ejemplo, si a la hora de intentar acceder a una página web determinada un alto porcentaje de las veces no está disponible, en términos profesionales nos da cierta inseguridad de cómo manejan sus aplicaciones, y en otros términos perderemos un considerable número de visitas e interés por los usuarios.

Si hay un campo en donde Linux he visto que ha destacado o es el jugador titular indiscutible, es en el Clustering. La capacidad de unir varias máquinas virtuales o físicas para dar servicio a la vez de forma que se balancean recursos para evitar sobrecargas de algunos servidores y en casos de que alguno falle y no pueda darlo. Esto es algo transparente al usuario final, pues tendremos una dirección virtual en donde se accede a todo el servicio del Cluster y luego los nodos que son los que realmente proporcionarán el servicio.

Le hemos echado un vistazo al Top 500 de las Supercomputadores, y observamos que el Sistema Operativo Linux es el más utilizado en los Supercomputadores.

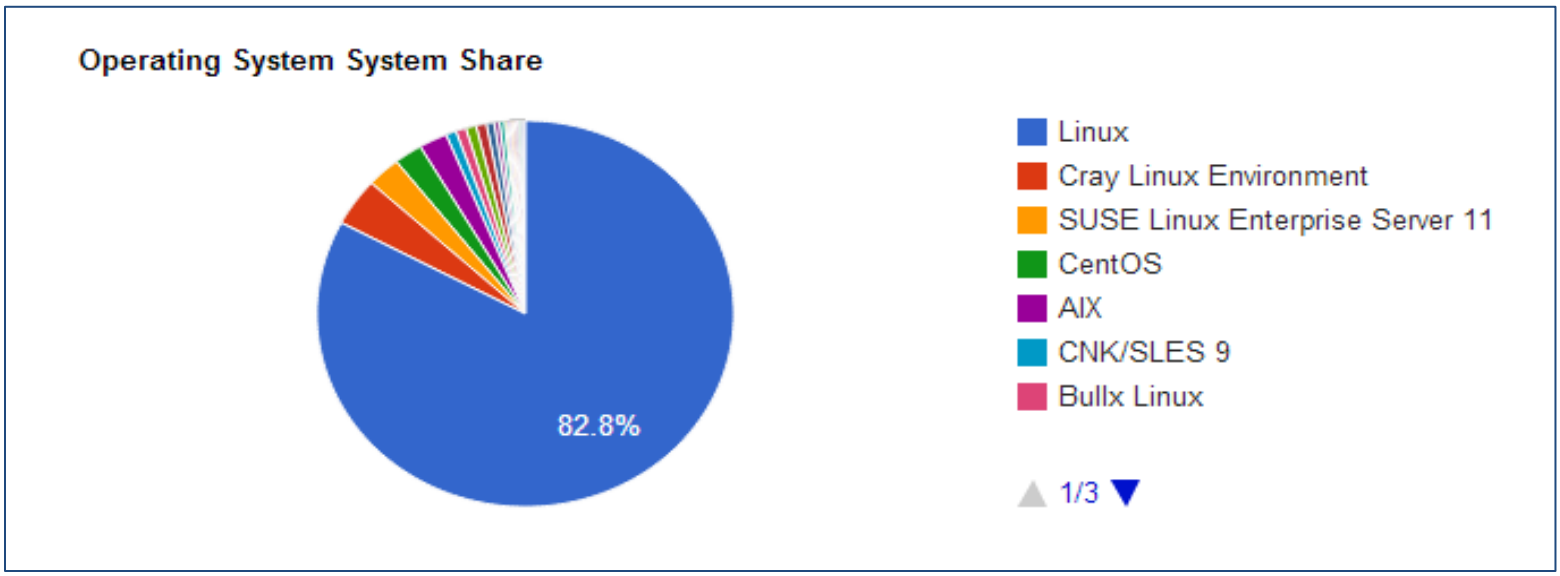


Figura 2.2.1

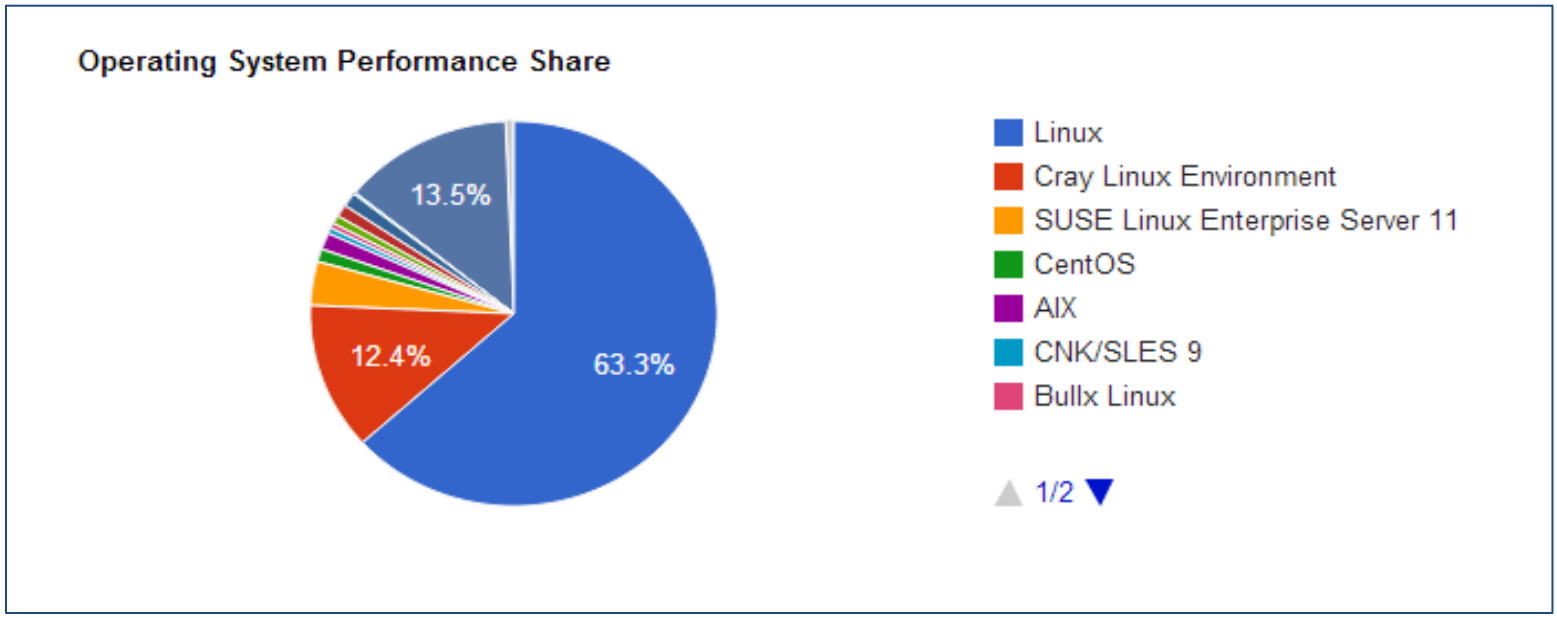


Figura 2.2.2

Operating System	Count	System Share (%)	Rmax (GFlops)	Rpeak (GFlops)	Cores
Linux	414	82.8	158,369,073	230,603,624	14,277,307
Cray Linux Environment	20	4	30,911,722	43,804,792	1,302,984
SUSE Linux Enterprise Server 11	13	2.6	9,174,795	13,081,620	432,150
CentOS	11	2.2	2,685,015	3,654,410	192,552
AIX	11	2.2	3,496,347	4,208,920	137,536
CNK/SLES 9	4	0.8	1,184,521	1,420,492	417,792
Bullx Linux	4	0.8	1,103,827	1,330,204	50,960
RHEL 6.2	4	0.8	1,738,900	2,132,582	102,528
Redhat Enterprise Linux 6	4	0.8	2,571,639	3,388,905	321,976
bullx SUperCOmputer Suite A.E.2.1	3	0.6	2,942,070	3,583,180	165,888
Redhat Linux	2	0.4	327,834	424,760	26,636
SLES10 + SGI ProPack 5	2	0.4	398,000	439,910	38,400
Windows Azure	1	0.2	151,300	167,731	8,064
CNL	1	0.2	165,600	201,216	20,960
Windows HPC 2008	1	0.2	180,600	233,472	30,720
Scientific Linux	1	0.2	188,725	199,680	9,600
RHEL 6.1	1	0.2	230,600	340,915	37,056
SUSE Linux	1	0.2	274,800	308,283	26,304
Kylin Linux	1	0.2	33,862,700	54,902,400	3,120,000
Super-UX	1	0.2	122,400	131,072	1,280

Figura 2.2.3

2.3 Sistemas Operativos.

En este apartado vamos a describir algunos de los Sistemas Operativos mayormente utilizados en el entorno de Servidores. Existe una amplia gama de sistemas operativos, los cuales poseen características distintas dependiendo del desarrollador.

A continuación nombramos algunos de los Sistemas Operativos para Servidores más relevantes y conocidos en la actualidad según el sitio web Distro Watch, poniendo énfasis en sus principales características.

2.3.1 Windows Server 2008.

Windows Server 2008 es el nombre de un sistema operativo de Microsoft diseñado para servidores.

Es el sucesor de Windows Server 2003, distribuido al público casi cinco años después. Al igual que Windows 7, Windows Server 2008 se basa en el núcleo Windows NT 6.1. Entre las mejoras de esta edición, se destacan nuevas funcionalidades para el Active Directory, nuevas prestaciones de virtualización y administración de sistemas, la inclusión de IIS 7.5 y el soporte para más de 256 procesadores. Hay siete ediciones diferentes: Foundation, Standard, Enterprise, Datacenter, Web Server, HPC Server y para Procesadores Itanium.

- **Características.**

Hay algunas diferencias (unas sutiles y otras no tanto) con respecto a la arquitectura del nuevo Windows Server 2008, que pueden cambiar drásticamente la manera en que se usa este sistema operativo. Estos cambios afectan a la manera en que se gestiona el sistema hasta el punto de que se puede llegar a controlar el hardware de forma más efectiva, se puede controlar mucho mejor de forma remota y cambiar de forma radical la política de seguridad. Entre las mejoras que se incluyen, están:

- Nuevo proceso de reparación de sistemas NTFS: proceso en segundo plano que repara los archivos dañados.
- Creación de sesiones de usuario en paralelo: reduce tiempos de espera en los Terminal Services y en la creación de sesiones de usuario a gran escala.
- Cierre limpio de Servicios.
- Sistema de archivos SMB2: de 30 a 40 veces más rápido el acceso a los servidores multimedia.
- Address Space Load Randomization (ASLR): protección contra malware en la carga de controladores en memoria.
- Windows Hardware Error Architecture (WHEA): protocolo mejorado y estandarizado de reporte de errores.
- Virtualización de Windows Server: mejoras en el rendimiento de la virtualización.
- PowerShell: inclusión de una consola mejorada con soporte GUI para administración.
- Server Core: el núcleo del sistema se ha renovado con muchas y nuevas mejoras.

- **Ediciones.**

La mayoría de las ediciones de Windows Server 2008 están disponibles en x86-64 (64 bits) y x86 (32 bits). Windows Server 2008 para sistemas basados en Itanium soporta procesadores IA-64. La versión IA-64 se ha optimizado para escenarios con altas cargas de trabajo como servidores de bases de datos y aplicaciones de línea de negocios (LOB). Por ende no está optimizado para su uso como servidor de archivos o servidor de medios. Microsoft ha anunciado que Windows Server 2008 será el último sistema operativo para servidores disponible en 32 bits. Windows Server 2008 está disponible en las ediciones que figuran a continuación, similar a Windows Server 2003.

- Windows Server 2008 Standard Edition (x86 y x86-64)
- Windows Server 2008 Todas las Ediciones (Solo 64Bit)
- Windows Server 2008 Enterprise Edition (x86 y x86-64)
- Windows Server 2008 Datacenter Edition (x86 y x86-64)
- Windows Server 2008 R2 Standard Edition (Solo 64Bit)
- Windows Server 2008 R2 Todas las Ediciones (Solo 64Bit)
- Windows Server 2008 R2 Enterprise Edition (Solo 64Bit)
- Windows Server 2008 R2 Datacenter Edition (Solo 64Bit)
- Windows HPC Server 2008 (reemplaza Windows Compute Cluster Server 2003)
- Windows Web Server 2008 R2 (Solo 64Bit)

- Windows Storage Server 2008 (x86 y x86-64)
- Windows Small Business Server 2008 (Nombre clave "Cougar") (x86-64) para pequeñas empresas
- Windows Essential Business Server 2008 (Nombre clave "Centro") (x86-64) para empresas de tamaño medio3
- Windows Server 2008 para sistemas basados en Itanium
- Windows Server 2008 R2 Foundation Server

Server Core está disponible en las ediciones Web, Standard, Enterprise y Datacenter, aunque no es posible usarla en la edición Itanium. Server Core es simplemente una opción de instalación alterna soportada y en sí no es una edición propiamente dicha. Cada arquitectura dispone de un DVD de instalación independiente.

Windows Server 2008 Standard Edition, y Windows Server 2008 R2 Standard Edition estaban disponibles gratuitamente para estudiantes a través del programa Microsoft DreamSpark.

Actualmente Windows Server 2008 Standard Edition (32 y 64bits), Windows Server 2008 Enterprise Edition (32 y 64bits), Windows Server 2008 Datacenter Edition (32 y 64bits), Windows Server 2008 R2 Standard Edition (con y sin SP1), Windows Server 2008 R2 Web Edition (con y sin SP1), Windows Server 2008 R2 Enterprise Edition (con y sin SP1), y Windows Server 2008 R2 Datacenter Edition (con y sin SP1), están disponible gratuitamente para estudiantes a través del programa Microsoft DreamSpark, al renovarse la licencia.

2.3.2 CentOS.

CentOS (Community ENTerprise Operating System) es una bifurcación a nivel binario de la distribución Linux Red Hat Enterprise Linux RHEL, compilado por voluntarios a partir del código fuente liberado por Red Hat.

Red Hat Enterprise Linux se compone de software libre y código abierto, pero se publica en formato binario usable (CD-ROM o DVD-ROM) solamente a suscriptores pagados. Como es requerido, Red Hat libera todo el código fuente del producto de forma pública bajo los términos de la Licencia pública general de GNU y otras licencias. Los desarrolladores de CentOS usan ese código fuente para crear un producto final que es muy similar al Red Hat Enterprise Linux y está libremente disponible para ser bajado y usado por el público, pero no es mantenido ni asistido por Red Hat. Existen algunos Clones de Red Hat Enterprise Linux.

CentOS usa yum para bajar e instalar las actualizaciones, herramienta también utilizada por Fedora.

En el año 2014 CentOS fue adquirido por Red Hat.

- **Requisitos de Sistema**

Hardware recomendado para operar:

Sin entorno de escritorio:

- Memoria RAM: 64 MB (mínimo).
- Espacio en Disco Duro: 1024 MB (mínimo) - 2 GB (recomendado).

Con entorno de escritorio:

- Memoria RAM: 2 GB (mínimo).
- Espacio en Disco Duro: 20 GB (mínimo) - 40 GB (recomendado).

- **Arquitecturas**

CentOS soporta casi las mismas arquitecturas que Red Hat Enterprise Linux:

- Intel x86-compatible (32 bit) (Intel Pentium/AMD64

2.3.3 Fedora.

Es una distribución Linux para propósitos generales basada en RPM, que se caracteriza por ser un sistema estable, la cual es mantenida gracias a una comunidad internacional de ingenieros, diseñadores gráficos y usuarios que informan de fallos y prueban nuevas tecnologías. Cuenta con el respaldo y la promoción de Red Hat.

El proyecto no busca sólo incluir software libre y de código abierto, sino ser el líder en ese ámbito tecnológico. Algo que hay que destacar es que los desarrolladores de Fedora prefieren hacer cambios en las fuentes originales en lugar de aplicar los parches específicos en su distribución, de esta forma se asegura que las actualizaciones estén disponibles para todas las variantes de Linux. Max Spevack en una entrevista afirmó que: "Hablar de Fedora es hablar del rápido progreso del software libre y de código abierto." Durante sus primeras 6 versiones se llamó Fedora Core, debido a que solo incluía los paquetes más importantes del sistema operativo.

La última versión es Fedora 20, puesta a disposición del público el 17 de diciembre de 2013.

De acuerdo a DistroWatch, Fedora es la cuarta distribución de Linux más popular, por detrás de Linux Mint, Mageia y Ubuntu.

- **Características:**

1. Distribución.

El Proyecto Fedora se distribuye en muchas formas diferentes:

- **Fedora DVD** - un DVD con todos los paquetes disponibles.
- **Medios Vivos (Live CD)** - imágenes de CD o DVD que también pueden ser instalados en unidades USB.
- **Imagen de CD o USB** - usado para ser instalado sobre HTTP, FTP o NFS.
- **Imagen de rescate en CD o USB** - usado si alguna parte del sistema ha fallado y requiere ser reparado. También permite instalaciones desde Internet.

También se distribuyen variantes personalizadas de Fedora, las cuales son llamadas Fedora spins. Éstas son construidas de un set de paquetes de software específico y tienen una combinación de software para satisfacer las necesidades de un usuario final determinado. Los Fedora spins son desarrollados por diferentes grupos especiales de Fedora.

Yum es el administrador de paquetes del sistema. Las interfaces gráficas, como el pirut y el pup, son provistos de la misma forma que el puplet, los cuales ofrecen notificaciones visuales en el panel cuando las actualizaciones están disponibles. *apt-rpm* es una alternativa a yum, y puede ser más familiar para personas que hayan usado anteriormente distribuciones como Ubuntu o Debian, donde *apt-get* es el administrador de paquetes predeterminado. Adicionalmente, repositorios extra pueden ser agregados al sistema y de esta forma paquetes que no están disponibles en Fedora pueden ser instalados.

2. Repositorios.

En las primeras 6 versiones había dos repositorios principales: El Fedora Core y el Fedora Extras. Fedora Core contenía todos los paquetes básicos que eran requeridos por el sistema operativo, así como otros que eran distribuidos con los CD o DVD de la instalación. Fedora Extras, el repositorio secundario que estaba incluido en Fedora Core 3 era mantenido por la comunidad y no estaba incluido en los discos de instalación. En ese entonces los repositorios eran:

- Core: en el cual se encuentran los paquetes esenciales.
- Extras: en el cual se encuentran los paquetes más utilizados o demandados.
- Updates: en el cual se encuentran las actualizaciones periódicas.

Antes de que Fedora 7 fuese liberada, había un cuarto repositorio llamado Fedora Legacy, el cual era mantenido por la comunidad y su objetivo era extender el ciclo de vida de versiones anteriores de Fedora o Red Hat que hayan sido dejadas de ser soportadas oficialmente. Fedora Legacy dejó de existir en diciembre de 2006.

Desde Fedora 7, los repositorios Core y Extras han sido fusionados, desde que la distribución abandonó el término Core de su nombre.

Actualmente, Fedora recomienda (o utiliza) únicamente aquellos repositorios que disponen de paquetes de software libre, o código abierto, sin problemas de patentes. Ejemplos de paquetes problemáticos a nivel de patentes son determinados códecs de audio, módulos NTFS o drivers de ATI y NVIDIA.

Junto con los repositorios fundamentales indicados con anterioridad, algunos de los repositorios más utilizados son Atrpms, Livna, FreshRPM, Dag, y Dries.

En el repositorio de Livna se encuentran aquellos paquetes que, aun siendo legales, únicamente pueden ser descargados por el usuario final, como códecs para MP3 y otros formatos. El resto de los repositorios indicados no clasifica los paquetes según su licencia, sino según su funcionalidad. Así mismo, existe la posibilidad de incompatibilidades entre repositorios, especialmente entre Livna y Atrpm, debido principalmente a que emplean diferentes opciones de compilación y por ello las dependencias pueden llegar a ser distintas.

La herramienta habitual, en Fedora, para interactuar con los repositorios a través de línea de comandos se denomina Yum; así mismo existe un entorno gráfico Yum denominado Pirut (para tareas de instalación y eliminación de paquetes) y Pup (para tareas de actualización de paquetes). Yum posee un front-end llamado Yumex.

3. Seguridad.

SELinux ("Security-Enhanced Linux") se destaca entre las características de seguridad de Fedora, pues implementa una gran variedad de políticas de seguridad, incluyendo control de acceso obligatorio (MAC "Mandatory Access Control"), a través de los Módulos de Seguridad de Linux que están en el núcleo Linux del sistema.

La distribución está liderando las distribuciones que incorporan SELinux, habiéndolo introducido en Fedora Core 2. Sin embargo lo desactivó como elemento predeterminado, pues alteraba radicalmente la forma en que el sistema operativo funcionaba. Posteriormente fue activado por defecto en Fedora Core 3 introduciendo una política menos estricta. Fedora también tiene métodos propios para prevenir la sobrecarga del buffer y la utilización de rootkits. La verificación del buffer en tiempo de compilación, “Exec Shield” y restricciones en como la memoria del núcleo en /dev/mem puede ser accedida ayudan a prevenir esto.

2.3.4 openSUSE.

OpenSUSE es el nombre de la distribución y proyecto libre auspiciado por SUSE Linux GmbH (una división independiente de The Attachmate Group) y AMD para el desarrollo y mantenimiento de un sistema operativo basado en Linux. Después de adquirir SUSE Linux en enero de 2004, Novell decidió lanzar SUSE Linux Professional como un proyecto completamente de código abierto, involucrando a la comunidad en el proceso de desarrollo. La versión inicial fue una versión beta de SUSE Linux 10.0, y la última versión estable es openSUSE 13.1.

- **Características.**

1. openSUSE comparte muchas características con SUSE Linux Enterprise, ofreciendo por ejemplo:
2. AppArmor: otorga permisos a las aplicaciones en función de cómo se ejecutan e interaccionan con el sistema.
3. YaST: una aplicación que openSUSE utiliza para administrar el sistema e instalar software.
4. Xen: software de virtualización.
5. KDE y GNOME.
6. Compiz: un escritorio 3D que corre sobre Xgl.

- **Requerimientos de sistema.**

OpenSUSE 13.1 brinda soporte completo para computadoras con procesadores 32-bit i586 y 64-bit x86-64, también está contemplado el soporte para la arquitectura ARM. El soporte de procesadores PowerPC (PPC) fue abandonado después de openSUSE 11.1. Los requerimientos mínimos de hardware son:

Procesador: Intel Pentium III o AMD Athlon a 500MHz, aunque se recomienda un Intel Pentium 4 a 2.4 GHz, el equivalente en AMD o superior. Todos los procesadores x86 de Intel y AMD con instrucciones de 64-bit están dentro de los requisitos recomendados, indistintamente de su velocidad de proceso o número de núcleos.

Memoria RAM: Dependiendo del entorno a usar, pueden requerirse como mínimo entre 512 MB a 1 GB de RAM, los requisitos recomendados ascienden a 2 GB o más de RAM.

Disco duro: 3 GB de espacio libre para una instalación mínima, 5 GB si se instala un entorno de escritorio (aunque se recomienda una partición con más espacio).

Resolución de pantalla de 800x600: (aunque se recomienda 1024x768 o mayor), estando soportadas casi la totalidad de las tarjetas gráficas e integradas, entre estas las más populares del mercado como AMD, NVIDIA, Intel y VIA.

Las especificaciones mínimas en realidad pueden diferir. Procesadores más antiguos que todavía pertenecen a la familia i586 se pueden utilizar, como el AMD K6-2, sobre todo cuando se quitan los archivos de lenguaje/traducción, la documentación, y no se usa el entorno gráfico X, pudiendo con estas configuraciones, hacer sistemas router basados en consola incluso con menos de 500 MB de espacio en disco. También la mayoría del trabajo en consola puede alcanzar con unos 128 MB RAM, pudiendo usar SWAP en situaciones de uso intenso y en cuanto a la resolución, si no se va a usar entorno gráfico, puede bastar con una resolución 640x480 que es la del estándar VGA e incluso pudiendo funcionar sin monitor, administrando el sistema vía remota.

2.3.5 Ubuntu.

Ubuntu es un sistema operativo basado en Linux y que se distribuye como software libre, el cual incluye su propio entorno de escritorio denominado Unity. Su nombre proviene de la ética homónima, en la que se habla de la existencia de uno mismo como cooperación de los demás.

Está orientado al usuario novel y promedio, con un fuerte enfoque en la facilidad de uso y en mejorar la experiencia de usuario. Está compuesto de múltiple software normalmente distribuido bajo una licencia libre o de código abierto. Estadísticas web sugieren que la cuota de mercado de Ubuntu dentro de las distribuciones Linux es, aproximadamente, del 49%, y con una tendencia a aumentar como servidor web. Y un importante incremento activo de 20 millones de usuarios para fines del 2011.

Su patrocinador, Canonical, es una compañía británica propiedad del empresario sudafricano Mark Shuttleworth. Ofrece el sistema de manera gratuita, y se financia por medio de servicios vinculados al sistema operativo y vendiendo soporte técnico. Además, al mantenerlo libre y gratuito, la empresa es capaz de aprovechar los desarrolladores de la comunidad para mejorar los componentes de su sistema operativo. Extraoficialmente, la comunidad de desarrolladores proporciona soporte para otras derivaciones de Ubuntu, con otros entornos gráficos, como Kubuntu, Xubuntu, Edubuntu, Ubuntu Studio, Mythbuntu, Ubuntu Gnome y Lubuntu.

Canonical, además de mantener Ubuntu, también provee de una versión orientada a servidores, Ubuntu Server, una versión para empresas, Ubuntu Business Desktop Remix, una para televisores, Ubuntu TV, y una para usar el escritorio desde teléfonos inteligentes, Ubuntu for Android.

Cada seis meses se publica una nueva versión de Ubuntu. Esta recibe soporte por parte de Canonical durante nueve meses por medio de actualizaciones de seguridad, parches para bugs críticos y actualizaciones menores de programas. Las versiones LTS (Long Term Support), que se liberan cada dos años, reciben soporte durante cinco años en los sistemas de escritorio y de servidor.

- **Características.**

En su última versión, Ubuntu soporta oficialmente dos arquitecturas de hardware en computadoras personales y servidores: 32-bit (x86) y 64-bit (x86_64). Sin embargo, extraoficialmente, Ubuntu ha sido portado a más arquitecturas: ARM, PowerPC, SPARC e IA-64.

A partir de la versión 9.04, se empezó a ofrecer soporte extraoficial para procesadores ARM, comúnmente usados en dispositivos móviles. Al igual que la mayoría de los sistemas de escritorio basados en Linux, Ubuntu es capaz de actualizar a la vez todas las aplicaciones instaladas en la máquina a través de repositorios. Ubuntu está siendo traducido a más de 130 idiomas, y cada usuario es capaz de colaborar voluntariamente a esta causa, a través de Internet.

1. Ubuntu y la comunidad:

Los usuarios pueden participar en el desarrollo de Ubuntu, escribiendo código, solucionando bugs, probando versiones inestables del sistema, etc. Además, en febrero de 2008 se puso en marcha el sitio Brainstorm que permite a los usuarios proponer sus ideas y votar las del resto. También se informa de las ideas propuestas que se están desarrollando o están previstas.

2. Software incluido:

Ubuntu posee una gran gama de aplicaciones para llevar a cabo tareas cotidianas, entretenimiento, desarrollo y aplicaciones para la configuración de todo el sistema. La interfaz predeterminada de Ubuntu es Unity y utiliza en conjunto las aplicaciones de GNOME. Existen otras versiones extraoficiales mantenidas por la comunidad, con diferentes escritorios, y pueden ser instalados independientemente del instalado por defecto en Ubuntu.

3. Aplicaciones de Ubuntu:

Ubuntu es conocido por su facilidad de uso y las aplicaciones orientadas al usuario final. Las principales aplicaciones que trae Ubuntu por defecto son: navegador web Mozilla Firefox, cliente de mensajería instantánea Empathy, cliente de correo Thunderbird, reproductor multimedia Totem, reproductor de música Rhythmbox, gestor y editor de fotos Shotwell, administrador de archivos Nautilus, cliente de BitTorrent Transmission, cliente de escritorio remoto Remmina, grabador de discos Brasero, suite ofimática LibreOffice, lector de documentos PDF Evince, editor de texto Gedit, cliente para sincronizar y respaldar archivos en línea Ubuntu One (desarrollada por Canonical), y la tienda de aplicaciones para instalar/eliminar/comprar aplicaciones Centro de software de Ubuntu (también desarrollada por Canonical).

4. Seguridad y accesibilidad

El sistema incluye funciones avanzadas de seguridad y entre sus políticas se encuentra el no activar, de forma predeterminada, procesos latentes al momento de instalarse. Por eso mismo, no hay un cortafuego predeterminado, ya que supuestamente no existen servicios que puedan atentar a la seguridad del sistema. Para labores o tareas administrativas en la línea de comandos incluye una herramienta llamada sudo (de las siglas en inglés de SwitchUser do), con la que se evita el uso del usuario administrador. Posee accesibilidad e internacionalización, de modo que el sistema esté disponible para tanta gente como sea posible. Desde la versión 5.04, se utiliza UTF-8 como codificación de caracteres predeterminado.

No sólo se relaciona con Debian por el uso del mismo formato de paquetes .deb. También tiene uniones con esa comunidad, aunque raramente contribuyendo con cualquier cambio directa e inmediatamente, o sólo anunciándolos. Esto sucede en los tiempos de lanzamiento.

La mayoría de los empaquetadores de Debian son los que realizan también la mayoría de los paquetes importantes de Ubuntu.

5. Organización del software

Ubuntu internamente divide todo el software en cuatro secciones, llamadas "componentes", para mostrar diferencias en licencias y la prioridad con la que se atienden los problemas que informen los usuarios. Estos componentes son: main, restricted, universe y multiverse.

Por defecto se instalan paquetes de los componentes main y restricted. Los paquetes del componente universe de Ubuntu generalmente se basan en los paquetes de la rama inestable (Sid) y en el repositorio experimental de Debian.

- **main:** contiene solamente los paquetes que cumplen los requisitos de la licencia de Ubuntu, y para los que hay soporte disponible por parte de su equipo. Éste está pensado para que incluya todo lo necesario para la mayoría de los sistemas Linux de uso general. Los paquetes de este componente poseen ayuda técnica garantizada y mejoras de seguridad oportunas.
- **restricted:** contiene paquetes soportados por los desarrolladores de Ubuntu debido a su importancia, pero que no está disponible bajo ningún tipo de licencia libre para incluir en main. En este lugar se incluyen los paquetes tales como los controladores propietarios de algunas tarjetas gráficas, como por ejemplo, los de ATI y NVIDIA. El nivel de la ayuda es más limitado que para main, puesto que los desarrolladores pueden no tener acceso al código fuente.
- **universe:** contiene una amplia gama de programas, que pueden o no tener una licencia restringida, pero que no recibe apoyo por parte del equipo de Ubuntu sino por parte de la comunidad. Esto permite que los usuarios instalen toda clase de programas en el sistema guardándolos en un lugar aparte de los paquetes soportados: main y restricted.
- **multiverse:** contiene los paquetes sin soporte debido a que no cumplen los requisitos de software libre.

- **LTS: Soporte técnico extendido.**

Cada 2 años se libera una versión con soporte técnico extendido a la que se añade la terminación LTS.

Esto significa que los lanzamientos LTS contarán con actualizaciones de seguridad de paquetes de software por un periodo de tiempo extendido. En versiones anteriores, era de 3 años en el entorno de escritorio y 5 años en el servidor por parte de Canonical LTD, a diferencia de los lanzamientos de cada 6 meses de Ubuntu que sólo cuentan con 9 meses de soporte (anteriormente 18 meses). Desde la versión 12.04 LTS (Precise Pangolin), el soporte es de 5 años en las dos versiones (Escritorio y Servidor).

La primera LTS fue la versión 6.06 de la cual se liberó una remasterización (la 6.06.1) para la edición de escritorio y dos remasterizaciones (6.06.1 y 6.06.2) para la edición servidor, ambas incluían actualizaciones de seguridad y corrección de errores. La segunda LTS fue la versión 8.04, de la cual ya va por la cuarta y última revisión de mantenimiento (la 8.04.4). La tercera LTS fue la versión 10.04, fue liberada en abril de 2010, y cuya última versión de mantenimiento fue la 10.04.4. La cuarta versión LTS que ha sido lanzada es la 12.04, que fue liberada en abril de 2012.

- **Requisitos de Instalación.**

Los requisitos mínimos "recomendados", teniendo en cuenta los efectos de escritorio, deberían permitir ejecutar una instalación de **Ubuntu Server 12.04 LTS.**

- Procesador x86 a 700 MHz.
- Memoria RAM de 512 Mb.
- Disco Duro de 5 GB (swap incluida).
- Tarjeta gráfica y monitor capaz de soportar una resolución de 1024x768.
- Lector de DVD o puerto USB.

Los efectos de escritorio, proporcionados por Compiz, se activan por defecto en las siguientes tarjetas gráficas:

- Intel (i915 o superior, excepto GMA 500, nombre en clave "Poulsbo").
- NVidia (con su controlador propietario o el controlar abierto incorporado Nouveau).
- ATI (a partir del modelo Radeon HD 2000 puede ser necesario el controlador propietario fglrx).
- Para una instalación óptima, y sobre todo si se dispone de más de 3 GiB de RAM, existe también una versión de Ubuntu para sistemas de 64 bits.

3. ENTORNO TECNOLOGICO

En este capítulo describimos los paquetes (software) que hemos empleado en nuestro proyecto. Hemos hecho uso de software libre para llevar a cabo nuestro cometido.

En concreto, en este apartado hacemos alusión de manera general y entendible de las tecnologías actualmente utilizadas, y que nos han servido de soporte para obtener los resultados deseados.

3.1. Programación Bash.

Bash (Bourne again shell) es un programa informático cuya función consiste en interpretar órdenes. Está basado en la shell de Unix y es compatible con POSIX.

Fue escrito para el proyecto GNU y es el intérprete de comandos por defecto en la mayoría de las distribuciones de GNU con Linux. Su nombre es un acrónimo de Bourne-Again Shell (otro shell bourne), haciendo un juego de palabras (born-again significa renacimiento) sobre el Bourne shell (sh), que fue uno de los primeros intérpretes importantes de Unix.

Hacia 1978 Bourne era el intérprete distribuido con la versión del sistema operativo Unix Versión 7. Stephen Bourne, por entonces investigador de los Laboratorios Bell, escribió la versión original de Bourne. Brian Fox escribió Bash en 1987. En 1990, Chet Ramey se convirtió en su principal desarrollador. Bash es el intérprete predeterminado en la mayoría de sistemas GNU/Linux, además de Mac OS X Tiger, y puede ejecutarse en la mayoría de los sistemas operativos tipo Unix. También se ha llevado a Microsoft Windows por el proyecto Cygwin.

- **Sintaxis de Bash**

La sintaxis de órdenes de Bash es un superconjunto de instrucciones basadas en la sintaxis del intérprete Bourne. La especificación definitiva de la sintaxis de órdenes de Bash, puede encontrarse en el Bash Reference Manual distribuido por el proyecto GNU. Esta sección destaca algunas de sus únicas características.

La mayoría de los shell scripts (guiones de intérprete de órdenes) Bourne pueden ejecutarse por Bash sin ningún cambio, con la excepción de aquellos guiones del intérprete de órdenes, o consola, Bourne que hacen referencia a variables especiales de Bourne o que utilizan una orden interna de Bourne. La sintaxis de órdenes de Bash incluye ideas tomadas desde el Korn Shell (ksh) y el C Shell (csh), como la edición de la línea de órdenes, el historial de órdenes, la pila de directorios, las variables $RANDOM y $PPID, y la sintaxis de substitución de órdenes POSIX: $(...). Cuando se utiliza como un intérprete de órdenes interactivo, Bash proporciona autocompletado de nombres de programas, nombres de archivos, nombres de variables, etc., cuando el usuario pulsa la tecla TAB.

3.2. Servidor DNS.

BIND (Berkeley Internet Name Domain, anteriormente: Berkeley Internet Name Daemon) es el servidor de DNS más comúnmente usado en Internet, especialmente en sistemas Unix, en los cuales es un estándar de facto. Es patrocinado por la Internet Systems Consortium. BIND fue creado originalmente por cuatro estudiantes de grado en la University of California, Berkeley y liberado por primera vez en el 4.3BSD. Paul Vixie comenzó a mantenerlo en 1988 mientras trabajaba para la DEC.

Una nueva versión de BIND (BIND 9) fue escrita desde cero en parte para superar las dificultades arquitectónicas presentes anteriormente para auditar el código en las primeras versiones de BIND, y también para incorporar DNSSEC (DNS Security Extensions). BIND 9 incluye entre otras características importantes: TSIG, notificación DNS, nsupdate, IPv6, rndc flush, vistas, procesamiento en paralelo, y una arquitectura mejorada en cuanto a portabilidad. Es comúnmente usado en sistemas GNU/Linux.

3.3. Servidor NFS.

El Network File System (Sistema de Archivos de Red), es un protocolo de nivel de aplicación, según el Modelo OSI. Es utilizado para sistemas de archivos distribuido en un entorno de red de computadoras de área local. Posibilita que distintos sistemas conectados a una misma red accedan a ficheros remotos como si se tratara de locales. Originalmente fue desarrollado en 1984 por Sun Microsystems, con el objetivo de que sea independiente de la máquina, el sistema operativo y el protocolo de transporte, esto fue posible gracias a que está implementado sobre los protocolos XDR (presentación) y ONC RPC (sesión). El protocolo NFS está incluido por defecto en los Sistemas Operativos UNIX y la mayoría de distribuciones Linux.

- **Características:**

- El sistema NFS está dividido al menos en dos partes principales: un servidor y uno o más clientes. Los clientes acceden de forma remota a los datos que se encuentran almacenados en el servidor.
- Las estaciones de trabajo locales utilizan menos espacio de disco debido a que los datos se encuentran centralizados en un único lugar pero pueden ser accedidos y modificados por varios usuarios, de tal forma que no es necesario replicar la información.
- Los usuarios no necesitan disponer de un directorio “home” en cada una de las máquinas de la organización. Los directorios “home” pueden crearse en el servidor de NFS para posteriormente poder acceder a ellos desde cualquier máquina a través de la infraestructura de red.

- También se pueden compartir a través de la red dispositivos de almacenamiento como disqueteras, CD-ROM y unidades ZIP. Esto puede reducir la inversión en dichos dispositivos y mejorar el aprovechamiento del hardware existente en la organización.

Todas las operaciones sobre ficheros son síncronas. Esto significa que la operación sólo retorna cuando el servidor ha completado todo el trabajo asociado para esa operación. En caso de una solicitud de escritura, el servidor escribirá físicamente los datos en el disco, y si es necesario, actualizará la estructura de directorios, antes de devolver una respuesta al cliente. Esto garantiza la integridad de los ficheros.

- **Versiones:**

Hay tres versiones de NFS actualmente en uso:

- La versión 2 de NFS (NFSv2), es la más antigua y está ampliamente soportada por muchos sistemas operativos.
- La versión 3 de NFS (NFSv3) tiene más características, incluyendo manejo de archivos de tamaño variable y mejores facilidades de informes de errores, pero no es completamente compatible con los clientes NFSv2.
- NFS versión 4 (NFSv4) incluye seguridad Kerberos, trabaja con cortafuegos, permite ACLs y utiliza operaciones con descripción del estado.

3.4. Servidor DHCP.

DHCP (sigla en inglés de Dynamic Host Configuration Protocol) es un protocolo de red que permite a los clientes de una red IP obtener sus parámetros de configuración automáticamente. Se trata de un protocolo de tipo cliente/servidor en el que generalmente un servidor posee una lista de direcciones IP dinámicas y las va asignando a los clientes conforme éstas van estando libres, sabiendo en todo momento quién ha estado en posesión de esa IP, cuánto tiempo la ha tenido y a quién se la ha asignado después. Este protocolo se publicó en octubre de 1993, y su implementación actual está en la RFC 2131. Para DHCPv6 se publica el RFC 3315.

- **Asignación de direcciones IP.**

Cada dirección IP debe configurarse manualmente en cada dispositivo y, si el dispositivo se mueve a otra subred, se debe configurar otra dirección IP diferente. El DHCP le permite al administrador supervisar y distribuir de forma centralizada las direcciones IP necesarias y, automáticamente, asignar y enviar una nueva IP si fuera el caso en el dispositivo es conectado en un lugar diferente de la red.

El protocolo DHCP incluye tres métodos de asignación de direcciones IP:

- **Asignación manual o estática:** Asigna una dirección IP a una máquina determinada. Se suele utilizar cuando se quiere controlar la asignación de dirección IP a cada cliente, y evitar, también, que se conecten clientes no identificados.
- **Asignación automática:** Asigna una dirección IP a una máquina cliente la primera vez que hace la solicitud al servidor DHCP y hasta que el cliente la libera. Se suele utilizar cuando el número de clientes no varía demasiado.
- **Asignación dinámica:** el único método que permite la reutilización dinámica de las direcciones IP. El administrador de la red determina un rango de direcciones IP y cada dispositivo conectado a la red está configurado para solicitar su dirección IP al servidor cuando la tarjeta de interfaz de red se inicializa. El procedimiento usa un concepto muy simple en un intervalo de tiempo controlable. Esto facilita la instalación de nuevas máquinas clientes a la red.

El DHCP es una alternativa a otros protocolos de gestión de direcciones IP de red, como el BOOTP (Bootstrap Protocol). DHCP es un protocolo más avanzado, pero ambos son los usados normalmente.

- **Parámetros configurables.**

Un servidor DHCP puede proveer de una configuración opcional al dispositivo cliente.

Lista de opciones configurables:

- Dirección del servidor DNS
- Nombre DNS
- Puerta de enlace de la dirección IP
- Dirección de Publicación Masiva (broadcast address)
- Máscara de subred
- Tiempo máximo de espera del ARP (Protocolo de Resolución de Direcciones según siglas en inglés)
- MTU (Unidad de Transferencia Máxima según siglas en inglés) para la interfaz
- Servidores NIS (Servicio de Información de Red según siglas en inglés)
- Dominios NIS
- Servidores NTP (Protocolo de Tiempo de Red según siglas en inglés)
- Servidor SMTP
- Servidor TFTP
- Nombre del servidor WINS

3.5. Servidor PXE.

Preboot eXecution Environment (PXE) (Entorno de Ejecución de Prearranque), es un entorno para arrancar e instalar el sistema operativo en ordenadores a través de una red, de manera independiente de los dispositivos de almacenamiento de datos disponibles (como discos duros) o de los sistemas operativos instalados.

PXE fue introducido como parte del framework Wired for Management por Intel y fue descrito en la especificación (versión 2.1) publicada por Intel y Systemsoft el 20 de septiembre de 1999. PXE utiliza varios protocolos de red como IP, UDP, DHCP y TFTP, y conceptos como Globally Unique Identifier (GUID), Universally Unique Identifier (UUID) y Universal Network Device Interface (UNDI).

El término cliente PXE sólo se refiere al papel que la máquina juega en el proceso de arranque mediante PXE. Un cliente PXE puede ser un servidor, un ordenador de mesa, portátil o cualquier otra máquina que esté equipada con código de arranque PXE.

- **Funcionamiento.**

El firmware del cliente trata de encontrar un servicio de redirección PXE en la red para recabar información sobre los servidores de arranque PXE disponibles. Tras analizar la respuesta, el firmware solicitará al servidor de arranque apropiado el file path de un network bootstrap program (NBP), lo descargará en la memoria RAM del ordenador mediante TFTP, probablemente lo verificará, y finalmente lo ejecutará. Si se utiliza un único NBP para todos los clientes PXE se puede especificar mediante BOOTP sin necesidad de un proxy DHCP, pero aún será necesario un servidor TFTP.

- **Disponibilidad.**

PXE fue diseñado para funcionar sobre diferentes arquitecturas. La versión 2.1 de la especificación asigna identificadores de arquitectura a seis tipos distintos de sistemas, incluyendo IA-64 y DEC Alpha. Aunque la especificación sólo soporta completamente IA-32. Intel incluyó PXE en la EFI para IA-64, creando un estándar de facto con esta implementación.

- **Protocolo.**

El protocolo PXE consiste en una combinación de los protocolos DHCP y TFTP con pequeñas modificaciones en ambos. DHCP es utilizado para localizar el servidor de arranque apropiado, con TFTP se descarga el programa inicial de bootstrap y archivos adicionales.

Para iniciar una sesión de arranque con PXE el firmware envía un paquete de tipo DHCPDISCOVER extendido con algunas opciones específicas de PXE al puerto 67/UDP (puerto estándar del servicio DHCP). Estas opciones indican que el firmware es capaz de manejar PXE, pero serán ignoradas por los servidores DHCP estándar.

- **Proxy DHCP:**

Si un servicio de redirección PXE (Proxy DHCP) recibe un paquete DHCPDISCOVER extendido, responde con un paquete de difusión DHCPOFFER extendido con opciones PXE al puerto 68/UDP. Este paquete se difundirá hasta que la mayoría de los clientes PXE se auto configuren mediante DHCP. Los clientes se identificarán con su GUID/UUID.

Un paquete DHCPOFFER extendido contiene:

- Un campo PXE Discovery Control para indicar si se debe utilizar Multicasting, Broadcasting, o Unicasting para contactar con los servidores de arranque PXE
- Una lista con las direcciones IP de los servidores de arranque PXE
- Un menú en el que cada entrada representa un servidor de arranque PXE
- Un prompt que indica al usuario que pulse [Tecla de función|<F8>]] para ver el menú de arranque
- Un tiempo de espera que lanza la primera opción del menú de arranque cuando expira

El servicio de proxy DHCP debe ejecutarse sobre el mismo servidor que el servicio estándar de DHCP. Puesto que ambos servicios no pueden compartir el puerto 67/UDP, el Proxy DHCP se ejecuta sobre el puerto 4011/UDP y espera que los paquetes DHCPDISCOVER extendidos de los clientes PXE sean paquetes DHCPREQUEST. El servicio estándar DHCP debe enviar una combinación especial de opciones PXE en su paquete DHCPOFFER, de forma que los clientes PXE sepan que deben buscar un proxy DHCP en el mismo servidor, en el puerto 4011/UDP.

- **Servidor de arranque:**

Para contactar con cualquier servidor de arranque PXE el firmware debe obtener una dirección IP y el resto de información de un único paquete DHCPOFFER extendido. Tras elegir el servidor de arranque PXE apropiado el firmware envía un paquete DHCPREQUEST extendido mediante multicast o unicast al puerto 4011/UDP o broadcast al puerto 67/UDP. Este paquete contiene el servidor de arranque PXE y la capa de arranque PXE, permitiendo ejecutar múltiples tipos de servidores de arranque mediante un único daemon (o programa) de arranque. El paquete DHCPREQUEST extendido también puede ser un paquete DHCPINFORM.

Si un servidor de arranque PXE recibe un paquete DHCPREQUEST extendido como el descrito anteriormente y si está configurado para el tipo de servidor de arranque PXE y la arquitectura de clientes solicitados, debe responder devolviendo un paquete DHCPACK extendido son opciones específicas de PXE.

El contenido más importante de un paquete DHCPACK extendido es:

- El file path completo para descargar el NBP vía TFTP
- El tipo de servidor de arranque PXE y la capa de arranque PXE
- La configuración multicast TFTP, si debe utilizarse multicast TFTP

Un servidor de arranque PXE debe soportar Boot Integrity Services (BIS). BIS permite al cliente PXE verificar los NBPs descargados mediante un archivo de checksum que es descargado desde el mismo servidor de arranque que el NBP.

- **Network bootstrap program:**

Tras recibir el paquete DHCPACK solicitado, el Network Bootstrap Program es descargado y ejecutado en la RAM del cliente. Tiene acceso a las APIs del firmware PXE (Pre-boot, UDP, TFTP, Universal Network Device Interface, UNDI).

3.6. Servicio dnsmasq.

Dnsmasq es un ligero servidor DNS, TFTP y DHCP. Su propósito es proveer servicios DNS y DHCP a una red de área local.

Dnsmasq acepta búsquedas DNS y las responde desde un pequeño caché local, o las reenvía hacia un servidor DNS real recursivo. Carga el contenido de /etc/hosts, de tal forma que nombres de hosts locales, los cuales no aparecen en el DNS mundial puedan ser resueltos. También responde a búsquedas DNS para hosts configurados vía DHCP.

El servidor DHCP dnsmasq incluye soporte para asignación de direcciones estáticas y redes múltiples. Automáticamente envía un predeterminado sensible de opciones DHCP, y puede ser configurado para enviar cualquier opción DHCP deseadas, incluyendo opciones encapsuladas por vendedores. Incluye un servidor seguro TFTP solo-lectura para permitir el inicio vía red/PXE de hosts DHCP. También incluye soporte para BOOTP. Dnsmasq incluye soporte IPv6 para DNS, pero no para DHCP.

Al inicio, dnsmasq lee /etc/dnsmasq.conf, si existe. En FreeBSD, el archivo es /usr/local/etc/dnsmasq.conf. El formato de este archivo consiste de una opción por línea, exactamente como las opciones largas detalladas en la sección OPCIONES pero sin el "--" al frente. Líneas que comienzan con # son comentarios y son ignoradas. Para opciones que solo pueden ser especificadas una sola vez, el archivo de configuración invalida la línea de comandos.

Al recibir un SIGHUP dnsmasq libera su cache y entonces recarga /etc/hosts y /etc/ethers al igual que cualquier archivo brindado con --dhcp-hostsfile, --dhcp-optsfile, o --addn-hosts. El archivo guion de cambio de arriendos es llamado para todos los arriendos DHCP existentes. Si --no-poll está fijado entonces SIGHUP también re-lee /etc/resolv.conf. SIGHUP NO re-lee el archivo de configuración.

Dnsmasq es un reenviador de búsquedas DNS: no puede responder búsquedas arbitrarias comenzando desde los servidores root pero reenvía dichas búsquedas a un servidor DNS recursivo, el cual es típicamente proveído por el proveedor de Internet. Por predeterminado, dnsmasq lee /etc/resolv.conf para descubrir las direcciones IP de los servidores DNS upstream que debe usar, dado a que esta información es normalmente almacenada allí. Amenos que --

no-poll sea usado, dnsmasq revisa el tiempo de modificación de /etc/resolv.conf (o equivalente si --resolv-file es usado) y lo relee si ha cambiado. Esto permite que servidores DNS sean fijados dinámicamente vía PPP o DHCP ya que ambos protocolos brindan esta información. La ausencia de /etc/resolv.conf no es un error ya que pudo haber sido creada antes de que una conexión PPP haya existido. Dnsmasq simplemente sigue revisando en caso de que /etc/resolv.conf sea creado en algún momento. A dnsmasq se le puede decir que revise más de un archivo resolv.conf. Esto es útil en una laptop, donde ambos PPP y DHCP podrían estar siendo usados: dnsmasq puede ser fijado para revisar ambos /etc/ppp/resolv.conf y /etc/dhcpc/resolv.conf y usará el contenido del que haya cambiado más recientemente, brindando así la habilidad de cambio automático entre servidores DNS.

El sistema de etiquetas funciona de la siguiente manera: Para cada pedido DHCP, dnsmasq colecciona un juego de etiquetas válidas de líneas de configuración activas que incluyen set:<tag>, incluyendo una del dhcp-range usado para alocar la dirección, una de cualquier dhcp-host que coincida (y "known" si un dhcp-host coincide). La etiqueta "bootp" es fijada para pedidos BOOTP, y una etiqueta cuyo nombre es el nombre de la interface donde llegó el pedido también es fijada.

Cualquier línea de configuración que incluya uno o más construcciones tag:<tag> solo será válida si todas las etiquetas coinciden en el juego derivado arriba. Típicamente esto es dhcp-option.

Dhcp-range puede tener un nombre de interface brindado como "interface:<interface-name>". La semántica de esto es así: Para DHCP, si cualquier otro dhcp-range existe _sin_ un nombre de interface, entonces el nombre de interface es ignorado y dnsmasq se comporta como si las partes de interface no existieran, de otra forma DHCP solo se provee a interfaces mencionadas en declaraciones dhcp-range. Para DNS, si no hay opciones --interface o --listen-address el comportamiento no se modifica por la parte de interface. Si cualquiera de estas opciones está presente, las interfaces mencionadas en dhcp-ranges son agregadas all juego que obtienen servicio DNS.

3.7. Servicio NIS.

Network Information Service (conocido por su acrónimo NIS, que en español significa Sistema de Información de Red), es el nombre de un protocolo de servicios de directorios cliente-servidor desarrollado por Sun Microsystems para el envío de datos de configuración en sistemas distribuidos tales como nombres de usuarios y hosts entre computadoras sobre una red.

NIS está basado en ONC RPC, y consta de un servidor, una biblioteca de la parte cliente, y varias herramientas de administración.

- **Implementaciones:**

Hoy NIS está disponible prácticamente en todas las distribuciones de Unix, e incluso existen implementaciones libres. BSD Net-2 publicó una que ha sido derivada de una implementación de referencia de dominio público donada por Sun. El código de la biblioteca de la parte cliente de esta versión existe en la libc de GNU/Linux desde hace mucho tiempo, y los programas de administración fueron portados a GNU/Linux por Swen Thümmler. Sin embargo, falta un servidor NIS a partir de la implementación de referencia.

Peter Eriksson ha desarrollado una implementación nueva llamada NYS. Soporta tanto NIS básico como la versión mejorada de Sun NIS+.1 NYS no sólo proporciona una serie de herramientas NIS y un servidor, sino que también añade un completo juego nuevo de funciones de biblioteca que necesita compilar en su libc si quiere utilizarlas. Esto incluye un esquema nuevo de configuración para la resolución de nombres de nodo que sustituye al esquema actual que usa el fichero "host.conf".

La libc de GNU, conocida como libc6 en la comunidad GNU/Linux, incluye una versión actualizada del soporte de NIS tradicional desarrollado por Thorsten Kukuk. Soporta todas las funciones de biblioteca que proporcionaba NYS, y también utiliza el esquema avanzado de configuración de NYS. Todavía se necesitan las herramientas y el servidor, pero utilizando la libc de GNU se ahorra el trabajo de tener que parchear y recompilar la biblioteca.

3.8. Condor.

Condor es un proyecto de la Universidad de Wisconsin-Madison (UWMadison). Está ideado para aprovechar al máximo la capacidad computacional de una red de ordenadores. Normalmente solo disponemos de la potencia del ordenador que estamos usando para ejecutar nuestros trabajos, y si, por ejemplo, tuviéramos que lanzar 100 veces un mismo programa con distinta entrada, tendríamos que hacerlo secuencialmente con la consecuente pérdida de tiempo. Condor nos permite ejecutar nuestro trabajo en tantas máquinas como haya disponibles, por lo que, en el mejor de los casos, nuestro trabajo finalizará en el tiempo que tarda en ejecutarse el más lento de nuestros procesos.

Condor pone a nuestra disposición toda la capacidad de cálculo desaprovechada en nuestra red, de esta manera, los recursos disponibles se incrementan considerablemente. Condor nos será útil siempre que necesitemos ejecutar un trabajo intenso, tanto computacionalmente como en el tiempo. Al aprovechar solamente recursos ociosos no incluye en el uso cotidiano de los ordenadores.

Además, nos permite:

- Conocer el estado de nuestros trabajos en cada momento.
- Implementar nuestras propias políticas de orden de ejecución.
- Mantener un registro de la actividad de nuestros trabajos.
- Añadir tolerancia a fallos a nuestros trabajos.

3.9. MPI.

MPI ("Message Passing Interface", Interfaz de Paso de Mensajes) es un estándar que define la sintaxis y la semántica de las funciones contenidas en una biblioteca de paso de mensajes diseñada para ser usada en programas que exploten la existencia de múltiples procesadores.

El paso de mensajes es una técnica empleada en programación concurrente para aportar sincronización entre procesos y permitir la exclusión mutua, de manera similar a como se hace con los semáforos, monitores, etc.

Su principal característica es que no precisa de memoria compartida, por lo que es muy importante en la programación de sistemas distribuidos.

Los elementos principales que intervienen en el paso de mensajes son el proceso que envía, el que recibe y el mensaje.

Dependiendo de si el proceso que envía el mensaje espera a que el mensaje sea recibido, se puede hablar de paso de mensajes síncrono o asíncrono. En el paso de mensajes asíncrono, el proceso que envía, no espera a que el mensaje sea recibido, y continúa su ejecución, siendo posible que vuelva a generar un nuevo mensaje y a enviarlo antes de que se haya recibido el anterior. Por este motivo se suelen emplear buzones, en los que se almacenan los mensajes a espera de que un proceso los reciba.

Generalmente empleando este sistema, el proceso que envía mensajes sólo se bloquea o para, cuando finaliza su ejecución, o si el buzón está lleno. En el paso de mensajes síncrono, el proceso que envía el mensaje espera a que un proceso lo reciba para continuar su ejecución. Por esto se suele llamar a esta técnica encuentro, o rendezvous. Dentro del paso de mensajes síncrono se engloba a la llamada a procedimiento remoto, muy popular en las arquitecturas cliente/servidor.

La Interfaz de Paso de Mensajes (conocido ampliamente como MPI, siglas en inglés de Message Passing Interface) es un protocolo de comunicación entre computadoras. Es el estándar para la comunicación entre los nodos que ejecutan un programa en un sistema de memoria distribuida. Las implementaciones en MPI consisten en un conjunto de bibliotecas de rutinas que pueden ser utilizadas en programas escritos en los lenguajes de programación C, C++, Fortran y Ada. La ventaja de MPI sobre otras bibliotecas de paso de mensajes, es que los programas que utilizan la biblioteca son portables (dado que MPI ha sido implementado para casi toda arquitectura de memoria distribuida), y rápidos, (porque cada implementación de la biblioteca ha sido optimizada para el hardware en la cual se ejecuta).

3.10. Almacenamiento RAID.

El acrónimo RAID (del inglés Redundant Array of Independent Disks, originalmente Redundant Array Inexpensive Disks), traducido como "conjunto redundante de discos independientes", hace referencia a un sistema de almacenamiento de datos que usa múltiples unidades de almacenamiento de datos (discos duros o SSD) entre los que se distribuyen o replican los datos. Dependiendo de su configuración (a la que suele llamarse "nivel"), los beneficios de un RAID respecto a un único disco son uno o varios de los siguientes: mayor integridad, mayor tolerancia a fallos, mayor Throughput (rendimiento) y mayor capacidad. En sus implementaciones originales, su ventaja clave era la habilidad de combinar varios dispositivos de bajo coste y tecnología más antigua en un conjunto que ofrecía mayor capacidad, fiabilidad, velocidad o una combinación de éstas que un solo dispositivo de última generación y coste más alto.

En el nivel más simple, un RAID combina varios discos duros en una sola unidad lógica. Así, en lugar de ver varios discos duros diferentes, el sistema operativo ve uno solo. Los RAIDs suelen usarse en servidores y normalmente (aunque no es necesario) se implementan con unidades de disco de la misma capacidad. Debido al decremento en el precio de los discos duros y la mayor disponibilidad de las opciones RAID incluidas en los chipsets de las placas base, los RAIDs se encuentran también como opción en las computadoras personales más avanzadas. Esto es especialmente frecuente en las computadoras dedicadas a tareas intensivas y que requiera asegurar la integridad de los datos en caso de fallo del sistema. Esta característica no está obviamente disponible en los sistemas RAID por software, que suelen presentar por tanto el problema de reconstruir el conjunto de discos cuando el sistema es reiniciado tras un fallo para asegurar la integridad de los datos. Por el contrario, los sistemas basados en software son mucho más flexibles (permitiendo, por ejemplo, construir RAID de particiones en lugar de discos completos y agrupar en un mismo RAID discos conectados en varias controladoras) y los basados en hardware añaden un punto de fallo más al sistema (la controladora RAID).

Todas las implementaciones pueden soportar el uso de uno o más discos de reserva (hot spare), unidades preinstaladas que pueden usarse inmediatamente (y casi siempre automáticamente) tras el fallo de un disco del RAID. Esto reduce el tiempo del período de reparación al acortar el tiempo de reconstrucción del RAID.

- **Niveles RAID estándar.**

Los niveles RAID más comúnmente usados son:

- RAID 0: Conjunto dividido.
- RAID 1: Conjunto en espejo.
- RAID 5: Conjunto dividido con paridad distribuida.

- **RAID 0 (Data Striping):**

Un RAID 0 (también llamado conjunto dividido, volumen dividido, volumen seccionado) distribuye los datos equitativamente entre dos o más discos sin información de paridad que proporcione redundancia. Es importante señalar que el RAID 0 no era uno de los niveles RAID originales y que no es redundante. El RAID 0 se usa normalmente para incrementar el rendimiento, aunque también puede utilizarse como forma de crear un pequeño número de grandes discos virtuales a partir de un gran número de pequeños discos físicos. Un RAID 0 puede ser creado con discos de diferentes tamaños, pero el espacio de almacenamiento añadido al conjunto estará limitado por el tamaño del disco más pequeño (por ejemplo, si un disco de 300 GB se divide con uno de 100 GB, el tamaño del conjunto resultante será sólo de 200 GB, ya que cada disco aporta 100GB). Una buena implementación de un RAID 0 dividirá las operaciones de lectura y escritura en bloques de igual tamaño, por lo que distribuirá la información equitativamente entre los dos discos. También es posible crear un RAID 0 con más de dos discos, si bien, la fiabilidad del conjunto será igual a la fiabilidad media de cada disco entre el número de discos del conjunto; es decir, la fiabilidad total -medida como MTTF o MTBF- es (aproximadamente) inversamente proporcional al número de discos del conjunto (pues para que el conjunto falle es suficiente con que lo haga cualquiera de sus discos).

- **RAID 1 (Mirroring):**

Un RAID 1 crea una copia exacta (o espejo) de un conjunto de datos en dos o más discos. Esto resulta útil cuando el rendimiento en lectura es más importante que la capacidad. Un conjunto RAID 1 sólo puede ser tan grande como el más pequeño de sus discos. Un RAID 1 clásico consiste en dos discos en espejo, lo que incrementa exponencialmente la fiabilidad respecto a un solo disco; es decir, la probabilidad de fallo del conjunto es igual al producto de las probabilidades de fallo de cada uno de los discos (pues para que el conjunto falle es necesario que lo hagan todos sus discos).

Adicionalmente, dado que todos los datos están en dos o más discos, con hardware habitualmente independiente, el rendimiento de lectura se incrementa aproximadamente como múltiplo lineal del número de copias; es decir, un RAID 1 puede estar leyendo simultáneamente dos datos diferentes en dos discos diferentes, por lo que su rendimiento se duplica. Para maximizar los beneficios sobre el rendimiento del RAID 1 se recomienda el uso de controladoras de disco independientes, una para cada disco (práctica que algunos denominan splitting o duplexing).

Como en el RAID 0, el tiempo medio de lectura se reduce, ya que los sectores a buscar pueden dividirse entre los discos, bajando el tiempo de búsqueda y subiendo la tasa de transferencia, con el único límite de la velocidad soportada por la controladora RAID. Sin embargo, muchas tarjetas RAID 1 IDE antiguas leen sólo de un disco de la pareja, por lo que su rendimiento es igual al de un único disco. Algunas implementaciones RAID 1 antiguas también leen de ambos discos simultáneamente y comparan los datos para detectar errores.

Al escribir, el conjunto se comporta como un único disco, dado que los datos deben ser escritos en todos los discos del RAID 1. Por tanto, el rendimiento no mejora.

El RAID 1 tiene muchas ventajas de administración. Por ejemplo, en algunos entornos 24/7, es posible "dividir el espejo": marcar un disco como inactivo, hacer una copia de seguridad de dicho disco y luego "reconstruir" el espejo. Esto requiere que la aplicación de gestión del conjunto soporte la recuperación de los datos del disco en el momento de la división. Este procedimiento es menos crítico que la presencia de una característica de snapshot en algunos sistemas de archivos, en la que se reserva algún espacio para los cambios, presentando una vista estática en un punto temporal dado del sistema de archivos. Alternativamente, un conjunto de discos puede ser almacenado de forma parecida a como se hace con las tradicionales cintas.

- **RAID 5:**

Un RAID 5 (también llamado distribuido con paridad) es una división de datos a nivel de bloques distribuyendo la información de paridad entre todos los discos miembros del conjunto. El RAID 5 ha logrado popularidad gracias a su bajo coste de redundancia. Generalmente, el RAID 5 se implementa con soporte hardware para el cálculo de la paridad. RAID 5 necesitará un mínimo de 3 discos para ser implementado.

En el gráfico de ejemplo anterior, una petición de lectura del bloque "A1" sería servida por el disco 0. Una petición de lectura simultánea del bloque "B1" tendría que esperar, pero una petición de lectura de "B2" podría atenderse concurrentemente ya que sería servida por el disco 1.

Cada vez que un bloque de datos se escribe en un RAID 5, se genera un bloque de paridad dentro de la misma división (stripe). Un bloque se compone a menudo de muchos sectores consecutivos de disco. Una serie de bloques (un bloque de cada uno de los discos del conjunto) recibe el nombre colectivo de división (stripe). Si otro bloque, o alguna porción de un bloque, es escrita en esa misma división, el bloque de paridad (o una parte del mismo) es recalculada y vuelta a escribir. El disco utilizado por el bloque de paridad está escalonado de una división a la siguiente, de ahí el término "bloques de paridad distribuidos". Las escrituras en un RAID 5 son costosas en términos de operaciones de disco y tráfico entre los discos y la controladora.

Los bloques de paridad no se leen en las operaciones de lectura de datos, ya que esto sería una sobrecarga innecesaria y disminuiría el rendimiento. Sin embargo, los bloques de paridad se leen cuando la lectura de un sector de datos provoca un error de CRC. En este caso, el sector en la misma posición relativa dentro de cada uno de los bloques de datos restantes en la división y dentro del bloque de paridad en la división se utiliza para reconstruir el sector erróneo. El error CRC se oculta así al resto del sistema. De la misma forma, si falla un disco del conjunto, los bloques de paridad de los restantes discos son combinados matemáticamente con los bloques de datos de los restantes discos para reconstruir los datos del disco que ha fallado "al vuelo".

Lo anterior se denomina a veces Modo Interino de Recuperación de Datos (Interim Data Recovery Mode). El sistema sabe que un disco ha fallado, pero sólo con el fin de que el sistema operativo pueda notificar al administrador que una unidad necesita ser reemplazada: las aplicaciones en ejecución siguen funcionando ajenas al fallo. Las lecturas y escrituras continúan normalmente en el conjunto de discos, aunque con alguna degradación de rendimiento. La diferencia entre el RAID 4 y el RAID 5 es que, en el Modo Interno de Recuperación de Datos, el RAID 5 puede ser ligeramente más rápido, debido a que, cuando el CRC y la paridad están en el disco que falló, los cálculos no tienen que realizarse, mientras que en el RAID 4, si uno de los discos de datos falla, los cálculos tienen que ser realizados en cada acceso.

El fallo de un segundo disco provoca la pérdida completa de los datos.

El número máximo de discos en un grupo de redundancia RAID 5 es teóricamente ilimitado, pero en la práctica es común limitar el número de unidades. Los inconvenientes de usar grupos de redundancia mayores son una mayor probabilidad de fallo simultáneo de dos discos, un mayor tiempo de reconstrucción y una mayor probabilidad de hallar un sector irrecuperable durante una reconstrucción. A medida que el número de discos en un conjunto RAID 5 crece, el MTBF (tiempo medio entre fallos) puede ser más bajo que el de un único disco. Esto sucede cuando la probabilidad de que falle un segundo disco en los N-1 discos restantes de un conjunto en el que ha fallado un disco en el tiempo necesario para detectar, reemplazar y recrear dicho disco es mayor que la probabilidad de fallo de un único disco. Una alternativa que proporciona una protección de paridad dual, permitiendo así mayor número de discos por grupo, es el RAID 6.

Algunos vendedores RAID evitan montar discos de los mismos lotes en un grupo de redundancia para minimizar la probabilidad de fallos simultáneos al principio y el final de su vida útil.

Las implementaciones RAID 5 presentan un rendimiento malo cuando se someten a cargas de trabajo que incluyen muchas escrituras más pequeñas que el tamaño de una división (stripe). Esto se debe a que la paridad debe ser actualizada para cada escritura, lo que exige realizar secuencias de lectura, modificación y escritura tanto para el bloque de datos como para el de paridad. Implementaciones más complejas incluyen a menudo cachés de escritura no volátiles para reducir este problema de rendimiento.

En el caso de un fallo del sistema cuando hay escrituras activas, la paridad de una división (stripe) puede quedar en un estado inconsistente con los datos. Si esto no se detecta y repara antes de que un disco o bloque falle, pueden perderse datos debido a que se usará una paridad incorrecta para reconstruir el bloque perdido en dicha división. Esta potencial vulnerabilidad se conoce a veces como “agujero de escritura”. Son comunes el uso de caché no volátiles y otras técnicas para reducir la probabilidad de ocurrencia de esta vulnerabilidad.

- **Niveles RAID anidados.**

Muchas controladoras permiten anidar niveles RAID, es decir, que un RAID pueda usarse como elemento básico de otro en lugar de discos físicos. Resulta instructivo pensar en estos conjuntos como capas dispuestas unas sobre otras, con los discos físicos en la inferior.

Los RAIDs anidados se indican normalmente uniendo en un solo número los correspondientes a los niveles RAID usados, añadiendo a veces un «+» entre ellos. Por ejemplo, el RAID 10 (o RAID 1+0) consiste conceptualmente en múltiples conjuntos de nivel 1 almacenados en discos físicos con un nivel 0 encima, agrupando los anteriores niveles 1. En el caso del RAID 0+1 se usa más esta forma que RAID 01 para evitar la confusión con el RAID 1. Sin embargo, cuando el conjunto de más alto nivel es un RAID 0 (como en el RAID 10 y en el RAID 50), la mayoría de los vendedores eligen omitir el "+", a pesar de que RAID 5+0 sea más informativo.

Al anidar niveles RAID, se suele combinar un nivel RAID que proporcione redundancia con un RAID 0 que aumenta el rendimiento. Con estas configuraciones es preferible tener el RAID 0 como nivel más alto y los conjuntos redundantes debajo, porque así será necesario reconstruir menos discos cuando uno falle. (Así, el RAID 10 es preferible al RAID 0+1 aunque las ventajas administrativas de "dividir el espejo" del RAID 1 se perderían.)

Los niveles RAID anidados más comúnmente usados son:

- RAID 0+1: Un espejo de divisiones.
- RAID 1+0: Una división de espejos.
- RAID 30: Una división de niveles RAID con paridad dedicada.
- RAID 100: Una división de una división de espejos.
- RAID 10+1: Un Espejo de espejos.

3.11. Servicio LVS.

Linux Virtual Server (LVS) es una solución para gestionar balance de carga en sistemas Linux. Es un proyecto de código abierto iniciado por Wensong Zhang en mayo de 1998. El objetivo es desarrollar un servidor Linux de alto rendimiento que proporcione buena escalabilidad, confiabilidad y robustez usando tecnología Clustering.

Actualmente, la labor principal del proyecto LVS es desarrollar un sistema IP avanzado de balanceo de carga por software (IPVS), balanceo de carga por software a nivel de aplicación y componentes para la gestión de Clusters.

- **IPVS:** sistema IP avanzado de balanceo de carga por software implementado en el propio núcleo Linux y ya incluido en las versiones 2.4 y 2.6.
- **KTCPVS:** implementa balanceo de carga a nivel de aplicación en el propio núcleo Linux. Actualmente está en desarrollo.

Los usuarios pueden usar las soluciones LVS para construir un sistema altamente escalable, donde se garantiza una alta disponibilidad de los servicios de red, como son servicios web, correo electrónico o VoIP.

Las soluciones LVS ya han sido implementadas en muchos entornos operacionales reales, incluidos los proyectos Wikimedia (enero de 2006).

3.12. Alta Disponibilidad: Corosync, Pacemaker y ldirectord.

El proyecto Linux-HA (Linux de Alta Disponibilidad) provee una solución Cluster de alta disponibilidad para Linux, FreeBSD, OpenBSD, Solaris y Mac OS X promoviendo fiabilidad, disponibilidad y servicialidad.

El producto principal del proyecto es Heartbeat, un programa de gestión de Clusters portable con licencia GPL para Clusters de alta disponibilidad. Sus más importantes características son:

- Máximo número de nodos no establecidos. Heartbeat puede ser usado tanto para Clusters grandes como Clusters de menor tamaño.
- Motorización de recursos: recursos pueden ser reiniciados o movidos a otro nodo en caso de fallo.
- Mecanismo de cercado para remover nodos fallidos en el Cluster.
- Gestión de recursos basado en directivas, inter-dependencia de recursos y restricciones
- Reglas basadas en el tiempo permiten diferentes directivas dependiendo del tiempo.
- Varios scripts de recursos (para Apache, DB2, Oracle, PostgreSQL, etc.) incluidos.
- GUI para configurar, controlar y monitorizar recursos y nodos.

Alan Robertson se inspiró en esta descripción y pensó que el quizás podría escribir algo del software para que el proyecto actuara como una especie de semilla de cristal inicial de modo a ayudar el autoarranque del proyecto. Él consiguió ejecutar el software inicial el 18 de marzo de 1998. Creó el portal web para el proyecto el 19 de octubre de 1998, y la primera versión del software fue liberada el 15 de noviembre de 1998. El primer cliente en producción de este software fue Rudy Pawul de ISO-INE. El portal web de ISO-INE entró en producción en el segundo semestre de 1999. En este punto, el proyecto estaba limitado a dos nodos y la semántica absorbida muy simple y ninguna monitorización de recursos.

Esto fue subsanado con versión 2 del software, el cual añadía Clusters de nodos, monitorización de recursos, dependencias y directivas. La versión 2.0.0 salió publicada el 29 de julio del 2005. Este release representaba otro hito importante ya que esta es la primera versión donde las contribuciones más grandes (en términos de tamaño de código) fueron hechas por la comunidad Linux-HA a mayores. Esta serie de lanzamientos trajo el proyecto a un nivel característico de paridad o superioridad con respecto al software comercial HA.

A partir de la distribución 2.1.3 de Heartbeat, se ha sustituido el código del gestor de recursos del Cluster (el CRM) por el componente pacemaker. Pacemaker constituye, por sí mismo, un proyecto independiente y no es una bifurcación del proyecto original de Linux-HA. El CRM que utilizan las nuevas distribuciones de Heartbeat forma parte de este nuevo proyecto y no volverá a distribuirse como parte del proyecto principal.

Los objetivos que se pretenden con esta decisión son, entre otros:

- Dar soporte, por igual, tanto a las pilas de Cluster OpenAIS como a Heartbeat.
- Desacoplar los ciclos de desarrollo de los dos proyectos.
- Mejorar y hacer más estables las interfaces.
- El proyecto Pacemaker aconseja la utilización de OpenAIS.

Ldirectord: (Linux Director Daemon) es un proceso que corre en background usado para monitorear y administrar servidores reales en un Cluster LVS.

Ldirectord supervisa el estado de los servidores reales, solicitando periódicamente una URL conocida y comprobando que la respuesta contiene una cadena esperada. Si un servicio falla en el servidor, el servidor es sacado del grupo de servidores reales y se reinserta una vez que entre de nuevo en línea.

3.13. Virtualización con Linux.

Virtualización es la creación, a través de software, de una versión virtual de algún recurso tecnológico, como puede ser una plataforma de hardware, un sistema operativo, un dispositivo de almacenamiento u otros recursos de red.

Dicho de otra manera, se refiere a la abstracción de los recursos de una computadora, llamada Hypervisor o VMM (Virtual Machine Monitor) que crea una capa de abstracción entre el hardware de la máquina física (host) y el sistema operativo de la máquina virtual (virtual machine, guest), dividiéndose el recurso en uno o más entornos de ejecución.

Esta capa de software (VMM) maneja, gestiona y arbitra los cuatro recursos principales de una computadora (CPU, Memoria, Dispositivos Periféricos y Conexiones de Red) y así podrá repartir dinámicamente dichos recursos entre todas las máquinas virtuales definidas en el computador central. Esto hace que se puedan tener varios ordenadores virtuales ejecutándose en el mismo ordenador físico.

Tal término es antiguo; se viene usando desde 1960, y ha sido aplicado a diferentes aspectos y ámbitos de la informática, desde sistemas computacionales completos, hasta capacidades o componentes individuales.

La virtualización se encarga de crear una interfaz externa que encapsula una implementación subyacente mediante la combinación de recursos en localizaciones físicas diferentes, o por medio de la simplificación del sistema de control. Un avanzado desarrollo de nuevas plataformas y tecnologías de virtualización ha hecho que en los últimos años se haya vuelto a prestar atención a este concepto.

La máquina virtual en general simula una plataforma de hardware autónoma incluyendo un sistema operativo completo que se ejecuta como si estuviera instalado. Típicamente varias máquinas virtuales operan en un computador central. Para que el sistema operativo "guest" funcione, la simulación debe ser lo suficientemente grande (siempre dependiendo del tipo de virtualización).

Existen diferentes formas de virtualización: es posible virtualizar el hardware de servidor, el software de servidor, virtualizar sesiones de usuario, virtualizar aplicaciones y también se pueden crear máquinas virtuales en una computadora de escritorio.

En nuestro Cluster vamos a emplear la herramienta conocida como VirtualBox, ya que este nos ofrece múltiples beneficios en cuanto a virtualización se refiere. A seguidas, una breve descripción acerca de VirtualBox.

- **VirtualBox.**

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

Entre los sistemas operativos soportados (en modo anfitrión) se encuentran GNU/Linux, Mac OS X, OS/2 Warp, Microsoft Windows, y Solaris/OpenSolaris, y dentro de ellos es posible virtualizar los sistemas operativos FreeBSD, GNU/Linux, OpenBSD, OS/2 Warp, Windows, Solaris, MS-DOS y muchos otros.

La aplicación fue inicialmente ofrecida bajo una licencia de software privativo, pero en enero de 2007, después de años de desarrollo, surgió VirtualBox OSE (Open Source Edition) bajo la licencia GPL 2. Actualmente existe la versión privativa Oracle VM VirtualBox, que es gratuita únicamente bajo uso personal o de evaluación, y está sujeta a la licencia de "Uso Personal y de Evaluación VirtualBox" (VirtualBox Personal Use and Evaluation License o PUEL) y la versión Open Source, VirtualBox OSE, que es software libre, sujeta a la licencia GPL.

VirtualBox ofrece algunas funcionalidades interesantes, como la ejecución de máquinas virtuales de forma remota, por medio del Remote Desktop Protocol (RDP), soporte iSCSI, aunque estas opciones no están disponibles en la versión OSE.

En cuanto a la emulación de hardware, los discos duros de los sistemas invitados son almacenados en los sistemas anfitriones como archivos individuales en un contenedor llamado Virtual Disk Image, incompatible con los demás softwares de virtualización.

Otra de las funciones que presenta es la de montar imágenes ISO como unidades virtuales ópticas de CD o DVD, o como un disquete.

Tiene un paquete de controladores que permiten aceleración en 3D, pantalla completa, hasta 4 placas PCI Ethernet (8 si se utiliza la línea de comandos para configurarlas), integración con teclado y ratón.

4. DESCRIPCIÓN DE LA SOLUCIÓN

En este capítulo detallamos de forma explícita los aspectos que hemos abordado para lograr obtener de manera satisfactoria los objetivos propuestos en este proyecto. En concreto, se describe cada una de las partes que componen nuestro proyecto y las tecnologías empleadas en el mismo. Hacemos referencia acerca de todos aquellos elementos que hacen posible la obtención de los objetivos planteados.

En este mismo orden, en este apartado detallamos la configuración hardware del Cluster, el procedimiento que hemos seguido para la instalación completa de cada uno de sus nodos, y las herramientas utilizadas para el mantenimiento y tareas de reparación. También, describimos los componentes que hacen posible obtener un correcto equilibrado de carga y alta disponibilidad de los servicios que ofrece nuestro Cluster. Por último, detallamos las herramientas que hacen posible que el Cluster ofrezca un servicio de máquinas virtuales para usuarios remotos.

Con todas las posibles plataformas entre las que podemos elegir, uno se puede preguntar por qué seleccionar Linux como Sistema Operativo (S.O.) en el que guardar nuestras aplicaciones críticas. Después de todo, siendo el Clustering un tema actual, cada vendedor tiene su propia implementación de software para Clustering. Los principales S.O. soportan el Clustering. Microsoft incluye aplicaciones para el Clustering directamente en su S.O. Windows Server. Sun Microsystems ofrece su tecnología High Performance Cluster para computación paralela, al igual que Sun Cluster para alta accesibilidad. Incluso Hewlett Packard, IBM y SG1 soportan soluciones para Clusters.

Entonces, ¿Por qué hemos tomado Linux como S.O. para nuestra solución? Esto se debe a que al incorporar Linux, nos beneficiamos de la característica de ser un código abierto. Aunque el método del código fuente abierto permanece como un modelo viable, las grandes empresas son las que reciben mayor beneficio de esta filosofía, y esto nos lleva a la conclusión de que el soporte está garantizado.

Una de las ventajas de Linux es que éste se ejecuta en casi cualquier plataforma imaginable. Y ha sido capaz de demostrar que puede hacer funcionar grandes supercomputadores y baterías de servidores al igual que ordenadores de escritorio. Aunque la disponibilidad de controladores para Linux no es tan abundante como en otros sistemas operativos, hay todavía una gran cantidad de hardware que funciona sin dificultad. Linux también soporta una gran cantidad de hardware anterior, permitiendo a ordenadores antiguos volver a funcionar. Los creadores de Linux incluso lo imaginan como el primer sistema operativo para dispositivos empotrados porque el núcleo puede ser modificado y adaptado para cubrir cualquier necesidad. Ningún otro sistema operativo permite este nivel de versatilidad. Es este método de computación modular lo que hace que Linux sea perfecto para los Clusters.

4.1. Configuración del Cluster.

- **Nodos del Cluster.**

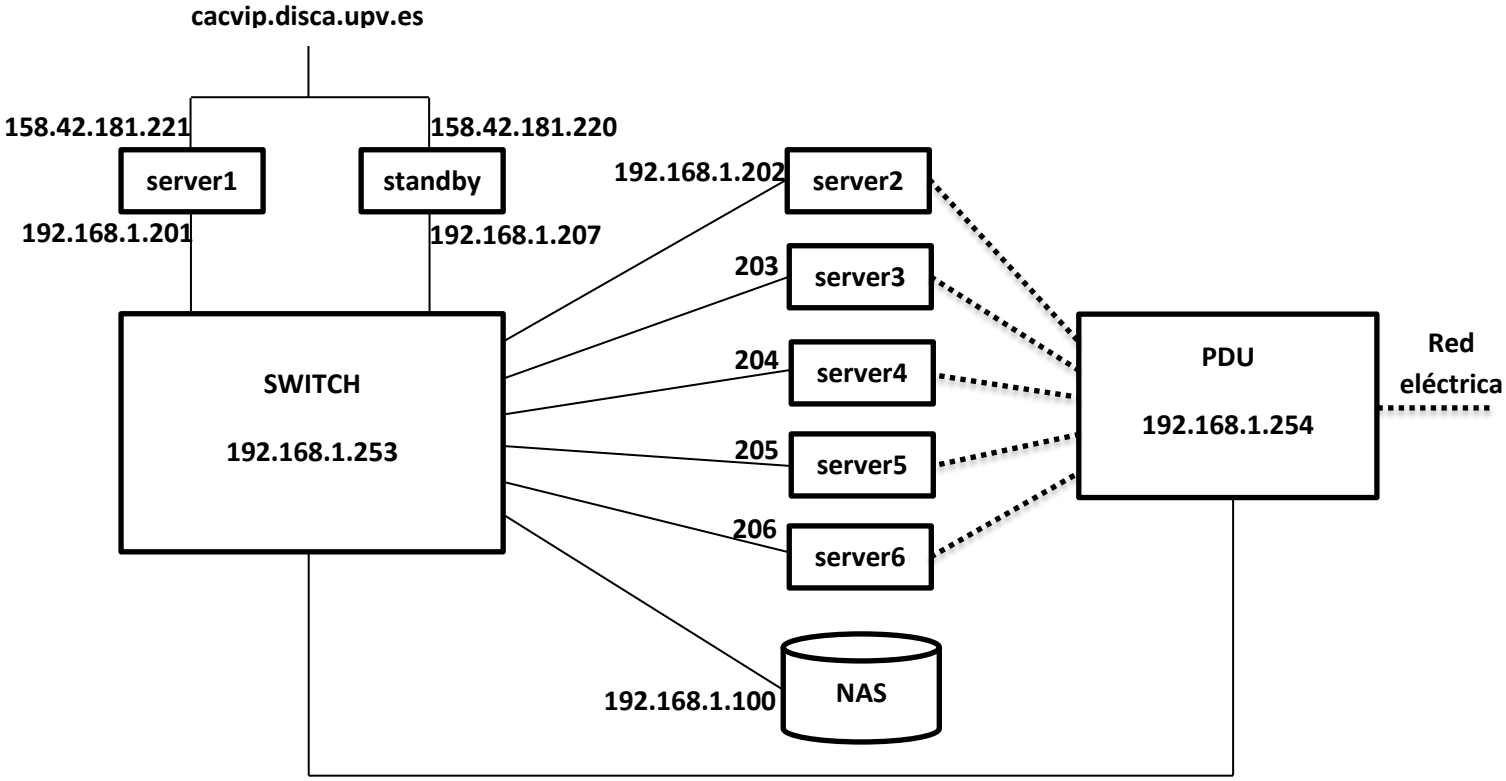
El Cluster utilizado en este proyecto está formado por 7 nodos **Bull Novascale R410** con las siguientes especificaciones:

1. **Procesador:** 1 Intel Xeon Quad-Core a 2.5GHz.
2. **Memoria RAM:** 2GB DDR2-800.
3. **Disco Duro:** 250GB SATA2.
4. **Interfaz de Red:** 10/100/1000 Fast Ethernet.
5. **Cabinet Desing:** Rack.
6. **Video:** On-board ATI ES1000 video controller with 16MB.
7. **Dimensiones externas:** 42,9 (alto) x 431,8 (ancho) × 508 mm (profundidad).
8. **Peso:** 9 Kg.
9. **Power supply:** 350W. Input voltages 110 to 220 V +/-10%. 50/60Hz +/- 1Hz.

Por asuntos de Alta Disponibilidad hemos realizado una segmentación de la siguiente manera:

1. **Nodo Master:** un nodo maestro. Conocido también como `server1`.
2. **Nodo Standby:** un nodo `Standby`. Que actúa como suplente del Master.
3. **Nodos de Cómputo (Esclavos):** cinco nodos esclavos. Su función principal es servir las páginas web del Cluster. También se encargan de servir las máquinas virtuales que aloja el Cluster. Estos son el `server2` al `server6`.

La siguiente figura muestra el diagrama de bloques de la máquina:



Como se puede observar, tanto el nodo Master como el Standby, están dotados de dos tarjetas de red. Una de ellas (eth1) permite la conexión con el exterior y nos proporciona acceso al Cluster. La otra interfaz de red (eth0) se encuentra conectada al resto de nodos mediante un conmutador de red (switch). Los nodos de cómputo (esclavos) se conectan entre sí a través de dicho conmutador mediante su interfaz de red (eth0).

En la siguiente tabla mostramos las direcciones IP y MAC Address que tienen configuradas cada uno de los nodos del Cluster.

	ETH0		ETH1			
NOMBRE	MAC	IP	MAC	IP		
SERVER1	00:15:17:27:B2:13	192.168.1.201	00:15:17:27:B2:15	158.42.181.221	MASTER	VIP
SERVER7	00:15:17:27:BF:C8	192.168.1.207	-	158.42.181.220	STANDBY	158.42.181.218
SERVER2	00:15:17:27:C3:8E	192.168.1.202	-	-		
SERVER3	00:15:17:27:C8:1D	192.168.1.203	-	-		
SERVER4	00:15:17:27:C3:58	192.168.1.204	-	-		
SERVER5	00:15:17:27:C3:AC	192.168.1.205	-	-		
SERVER6	00:15:17:27:BF:86	192.168.1.206	-	-		
NAS	00:00:00:00:00:00	192.168.1.100	-	-		

Los discos de todos los nodos se encuentran particionados de manera tal que pueda coexistir un sistema principal, con otro de pruebas. En particular, las particiones de los nodos son las siguientes:

- /dev/sda1 swap
- /dev/sda8 Sistema Principal
- /dev/sda9 Pruebas
- /dev/sda10 Auxiliar

Esta distribución de las particiones las hemos realizado así por motivos de tolerancia a fallos. Es decir, podemos iniciar el sistema tanto en la partición principal como en la auxiliar, pudiendo de esta manera instalar o reparar una partición dañada.

- **Sistema de Almacenamiento.**

El sistema de almacenamiento incorporado en el Cluster es un NAS (Network Attached Storage) **NAS Deluxe 2800R**, con las siguientes características técnicas:

- **Espacio de almacenamiento:** 7,5 TB accesible mediante NFS.
- **SATA Device:** 8 discos duros.
- **LAN Interface:** 10/100/1000 BASE-TX Auto MDI/MDI-X WOL supported.
- **Power Supply:** Redundant Power Supply.
- **RAID modes:** RAID 0, 1, 5, 6, 10, JBOD.
- **File Protocols:** SMB/CIFS, HTTP/HTTPS, FTP, NFS, AFP.

La alimentación eléctrica se aplica a cada nodo mediante una Unidad de Distribución de Potencia (PDU). Un conmutador KVM, una consola, el gabinete (rack) y los cables de conexión completan el sistema.

Mediante el conmutador KVM podemos seleccionar el nodo al que se conecta la consola extraíble. Esto puede hacerse manipulando el pulsador del KVM o mediante la pulsación secuencial de las teclas <Bloq despl.> <Bloq despl.> <Barra espaciadora>. Seguidamente con las teclas del cursor seleccionamos la máquina a acceder.

La siguiente tabla muestra el peso (Kg), la potencia eléctrica (W) y el precio (€) total de nuestro Cluster.

	Unidad	Total
Peso	9 Kg	63 Kg
Potencia Electrica	350 W	2450 W
Precio	€ 775,86	€ 5.431,02

4.2. Instalación del Sistema Operativo en el Cluster.

En este apartado se describe el procedimiento que hemos seguido para obtener una instalación efectiva del sistema operativo en la **Partición Principal (/dev/sda8)** y en la **Partición Auxiliar (/dev/sda10)**. También se detalla la instalación de los paquetes NIS, Condor y MPI, los cuales nos permiten la ejecución de tareas específicas en nuestro sistema.

La distribución Linux elegida para la instalación es la **Ubuntu 12.04 LTS Server Edition**, la cual resulta bastante conveniente por las múltiples ventajas que ofrece, así como también, gran flexibilidad de configuración.

Un aspecto importante a resaltar en este proceso de instalación, consiste en la utilización de herramientas (*scripts*) que permiten instalar los sistemas y aplicaciones (paquetes) de manera desatendida y automática. Esto resulta ventajoso en el sentido de que, como tenemos 7 nodos, sería bastante tedioso instalar uno por uno.

- **Proceso de Instalación.**

De manera general, el proceso de instalación que hemos seguido consiste en lo siguiente:

1. Instalación del NFS.
2. Instalación del Sistema Operativo en la Partición Principal (/dev/sda8) del nodo Master.
3. Instalación, configuración y puesta en marcha del Servidor PXE en el nodo Master.
4. Inicio de los servidores Esclavos (server2 al server6) mediante la red (PXE).
5. Clonación de la partición principal del nodo Master en la partición principal de los nodos Esclavos.
6. Configuración adicional en el nodo Master y los nodos Esclavos.
7. Apagado del Servidor PXE y arranque del servidor en modo DHCP.
8. Reinicio de los nodos Esclavos.

- **Herramientas utilitarias de instalación.**

Como paso previo a la instalación del sistema vamos a describir las herramientas utilitarias que nos permiten realizar tareas de configuración de una forma más automatizada y eficiente. Para facilitar la instalación, configuración y mantenimiento de los nodos de cómputo del Cluster, se han diseñado varias herramientas (`scripts`) que nos permiten lanzar órdenes y copiar ficheros mediante ssh desde el nodo Master hacia los nodos servidores. También, se ha creado una herramienta que genera las claves públicas para poder acceder a los servidores esclavos sin necesidad de colocar nuestra contraseña cada vez que ejecutemos una orden remota.

En concreto, estas herramientas utilitarias son las siguientes:

1. **Script psh:** este envía una orden mediante *ssh* a todos los servidores del Cluster.
2. **Script pscp:** se encarga de copiar un fichero desde el Master a los nodos esclavos del Cluster.
3. **Script keyscan.sh:** este script se encarga de generar las claves públicas de los nodos del Cluster.

Tenemos predefina una variable de entorno llamada CLUSTER_SIZE con el número de nodos del Cluster en su contenido.

```
#!/bin/sh

#Script psh. Envia una orden a todos los nodos del Cluster TFM.
i=2
while [ $i -le $CLUSTER_SIZE ]
  do
    #Para imprimir por pantalla el nodo y la orden que estamos pasando.
    echo server$i $1;
    #Nos conectamos al nodo y le pasamos la orden que escribo en el terminal.
    ssh server$i $1;
    let i=i+1
done
sleep 1s
echo "Operacion completada satisfactoriamente!!"
echo "*******+*******+*******+*******"
echo "Fin del script psh!!...*******"
```

Script psh

```
#!/bin/sh

#Script pscp. Para copiar un fichero en todos los nodos del Cluster TFM.
i=2
while [ $i -le $CLUSTER_SIZE ]
  do
    echo server$i $1;
    #Copia el fichero que se pasa en el primer argumento ($1),
    #en la ruta que pasa  mos como segundo argumento ($2).
    scp $1 server$i:$2;
    let i=i+1
done
sleep 1s
echo "Operacion completada satisfactoriamente!!"
echo "*******+*******+*******+*******"
echo "Fin del script pscp...*******"
echo "*******+*******+*******+*******"
```

Script pscp

```
#!/bin/bash

i=2
while [ $i -le $CLUSTER_SIZE ]
  do
    ssh-keyscan cluster$i >> /root/.ssh/known_hosts
    let i=i+1
  done
```

Script keyscan.sh

- **Instalación del NFS.**

Como se ha mencionado anteriormente en el apartado de Configuración del Cluster, tenemos incorporado en el Cluster un sistema de almacenamiento NAS (Network Attached Storage) **NAS Deluxe 2800R**. Este dispositivo ya se encuentra instalado y configurado por el equipo técnico del Departamento del DISCA de la Universidad Politécnica de Valencia (UPV). Este sistema ofrece una interfaz web donde podemos configurar los RAID de disco según soporta el NAS. Hemos verificado que el NAS exporta a sus clientes un RAID 0 en un directorio nombrado como */nfs*.

A continuación mostramos la configuración del fichero /etc/fstab donde se configura el NAS mediante NFS:

```
192.168.1.100:/raid0/data/lv0001 /nfs nfs nfsvers=3,auto,rw,
wsize=8192,rsize=8192      0     0
```

- **Instalación en el Nodo Master.**

En primer lugar, realizamos la instalación completa del nodo Master partiendo desde un **CD de instalación**. Esto así, ya que a partir de este clonaremos los nodos esclavos. La estrategia radica en realizar una única instalación y clonarla o replicarla en todos los nodos servidores que tenemos disponible.

El proceso que hemos seguido consiste en la instalación del sistema en el nodo Master tal y como queremos que quede en los nodos servidores y luego crear una imagen del sistema, generando así el nodo **"modelo"** a clonar.

Luego de finalizada la instalación del Sistema Operativo, el siguiente paso es preparar la imagen que vamos a clonar en los servidores esclavos. Para lograrlo, instalamos un servidor NFS para conseguir exportar una partición del sistema que sirva de punto de arranque para los nodos esclavos.

Instalamos el `grub` del sistema, modificamos convenientemente el fichero `/etc/hosts`, el cual contiene todos los nodos del Cluster con su correspondiente dirección IP. También, se ha modificado el fichero `/etc/network/interfaces`, ya que nuestro nodo Master posee dos interfaces de red, mientras que los nodos esclavos solamente poseen una.

Realizamos una copia de nuestro sistema en la partición exportada por NFS empleando el script *copyNFS*, el cual se encarga de copiar el sistema y realizar modificaciones en los ficheros `/etc/fstab` y `/etc/network/interfaces`. A continuación se muestra el contenido del script *copyNFS*:

```
#!/bin/sh

#Copia del sistema principal instalado al NFS.

cp -ax / /nfs/srv/

#Configuracion del fichero /nfs/srv/etc/fstab:

rm -r /nfs/srv/etc/fstab

echo "/dev/nfs       /       nfs defaults    0   0" >> /nfs/srv/etc/fstab

#Configuracion del fichero /nfs/srv/etc/network/interfaces:

rm -r /nfs/srv/etc/network/interfaces

echo "auto lo" >> /nfs/srv/etc/network/interfaces
echo "iface lo inet loopback" >> /nfs/srv/etc/network/interfaces
echo "####" >> /nfs/srv/etc/network/interfaces
echo "####" >> /nfs/srv/etc/network/interfaces
echo "iface eth0 inet dhcp" >> /nfs/srv/etc/network/interfaces

#Eliminar fichero corosync.

rm -r /nfs/srv/etc/default/corosync

echo "*******+*******+*******+*******"
echo "Configuracion completada satisfactoriamente!!*******"
echo "*******+*******+*******+*******"
echo "FIN DEL SCRPT copyNFS.sh*******"
```

Script *copyNFS.sh*

Luego de esto, instalamos el servicio DNS, empleando el paquete `bind9`, y habilitamos al nodo Master para que pueda dar acceso al exterior desde los nodos servidores modificando la tabla de NAT. Esto se consigue modificando el fichero `/etc/rc.local` en el cual se especifica que deseamos habilitar el ip_forward.

```
ip_forward=1
```

- **Instalación y configuración del Servidor PXE.**

Hemos puesto en marcha en el Nodo Master un **Servidor PXE** para que los nodos esclavos arranquen por red a través de este. Una vez hayan arrancado los nodos servidores, procedemos a replicar el sistema en cada una de las particiones primarias (partición principal), empleando un script de instalación denominado *servCLONE.sh*.

Para la instalación del servidor PXE en el Master hemos empleado dos scripts de instalación y configuración respectivamente. El script de instalación, denominado *configPXE-1.sh*, tiene la finalidad de instalar el paquete `syslinux (PXE)` y configurar convenientemente el fichero `/boot/pxelinux.cfg` para lograr nuestro objetivo en concreto, que los nodos a clonar puedan arrancar mediante red, específicamente, mediante el servicio de NFS.

```
#!/bin/sh

#Instalacion del servidor PXE.

apt-get install syslinux

cp /usr/lib/syslinux/pxelinux.0 /boot/

#Configuracion del servidor PXE.

mkdir /boot/pxelinux.cfg

echo "DEFAULT linux" > /boot/pxelinux.cfg/default
echo "LABEL linux" >> /boot/pxelinux.cfg/default
echo "KERNEL vmlinuz" >> /boot/pxelinux.cfg/default
echo "APPEND root=/dev/nfs initrd=/initrd_netboot nfsroot=192.168.1.201:/nfs/srv rw" >> /boot/pxelinux.cfg/default

echo "*******+*******+*******+*******"
echo "Configuracion completada satisfactoriamente!!"
echo "*******+*******+*******+*******"
sleep 2s
echo "Fin del script configPXE.sh!!*******"
```

Script *configPXE-1.sh*

Para la preparación y configuración del servidor PXE hemos utilizado el script *configPXE-2.sh*, el cual tiene la función de instalar el paquete `dnsmasq` y crear dos fichero de configuración. Uno con la habilidad de proveer al Master del servicio **dhcp y PXE**, y el otro habilitado para proporcionar al Master solo el servicio de **dhcp.**

A seguidas mostramos el script configPXE-2.sh.

```
#!/bin/sh

#Preparacion del servidor PXE.

apt-get install dnsmasq
sleep2s

/etc/init.d/dnsmasq stop
sleep 2s

#Fichero de configuracion /etc/dnsmasq-dhcp-pxe.conf
echo "port=0" >> /etc/dnsmasq-dhcp-pxe.conf
echo "enable-tftp" >> /etc/dnsmasq-dhcp-pxe.conf
echo "tftp-root=/boot" >> /etc/dnsmasq-dhcp-pxe.conf
echo "dhcp-boot=/pxelinux.0" >> /etc/dnsmasq-dhcp-pxe.conf
echo "dhcp-range=192.168.1.202,192.168.1.210,infinite" >> /etc/dnsmasq-dhcp-pxe.conf
echo "log-dhcp" >> /etc/dnsmasq-dhcp-pxe.conf
echo "dhcp-option=mtu,9000" >> /etc/dnsmasq-dhcp-pxe.conf
echo "read-ethers" >> /etc/dnsmasq-dhcp-pxe.conf

#Fichero de configuracion /etc/dnsmasq-dhcp.conf
echo "port=0" >> /etc/dnsmasq-dhcp.conf
echo "dhcp-range=192.168.1.202,192.168.1.210,infinite" >> /etc/dnsmasq-dhcp.conf
echo "log-dhcp" >> /etc/dnsmasq-dhcp.conf
echo "dhcp-option=mtu,9000" >> /etc/dnsmasq-dhcp.conf
echo "read-ethers" >> /etc/dnsmasq-dhcp.conf

echo "Configuracion completada satisfactoriamente!!"
echo "*******+*******+*******+*******"
sleep 1s
echo "Fin del script configPXE-2.sh!!*******"
```

Script *configPXE-2.sh*

- **Clonación a los Nodos Servidores.**

Para clonar efectivamente los nodos servidores del Cluster, hemos empleado dos scripts de instalación que se encargan de la clonación automática del sistema raíz del Nodo Master, a las correspondientes particiones principales (partición principal) de los nodos esclavos.

Una vez ya arrancados los nodos esclavos mediante el servidor PXE, procedemos a ejecutar el script *servCLONE.sh*, el cual tiene como finalidad preparar las particiones e instalar debidamente el sistema desde el servidor NFS.

Luego de que termine la instalación en cada uno de los nodos, ejecutamos el script *configNODE*. Este prepara los nodos para que cuando se reinicien puedan arrancar desde su **sistema principal**. Se encarga de apagar el servidor PXE en el Master, y habilitarlo para que solo ofrezca el servidor DHCP. También, realiza cambios en los ficheros `/etc/fstab` y `/etc/network/interfaces`, ya que estos son una copia del servidor principal.

```
#!/bin/sh

#Clonacion del sistema a los nodos servidores.

#Particionado de los discos de los servidores.
sfdisk -d /dev/sda > /nfs/srv/root/sda.out
sleep 1s

#Creacion tabla de particiones.
./psh "sfdisk -f /dev/sda < /root/sda.out"
sleep 1s

#Formatear la particion de arranque.
./psh "mkfs /dev/sda8"
sleep 1s

#Preparacion de la swap.
./psh "mkswap /dev/sda1"
./psh "swapon /dev/sda1"
sleep 1s

#Clonacion del sistema a los nodos servidores.
./psh "mkdir /mnt/sda8"
./psh "mount /dev/sda8 /mnt/sda8"
./psh "cp -ax / /mnt/sda8"
sleep 2s

echo "*******+*******+*******+********"
echo "Configuracion completada satisfactoriamente!!*******"
echo "*******+*******+*******+********"
echo "Fin del script servCLONE.sh...*******"
```

Script *servCLONE.sh*

```
#!/bin/sh

#Configuracion adicional de los nodos servidores y del nodo maestro.

#Configuracion del fichero /etc/fstab para los nodos servidores.
echo "/dev/sda8      /    ext4     defaults     0    0" >> /root/fstab
echo "/dev/sda1      none     swap     auto,defaults    0    0" >> /root/fstab

#Copia del fichero /root/fstab en los nodos servidores.
./pscp /root/fstab /mnt/sda8/etc

#Configuracion del fichero /etc/network/interfaces de los nodos servidores.
echo "auto lo" >> /root/interfaces
echo "iface lo inet loopback" >> /root/interfaces
echo "auto eth0" >> /root/interfaces
echo "iface eth0 inet dhcp" >> /root/interfaces

#Copia del fichero interfaces en los nodos servidores.
./pscp /root/interfaces /mnt/sda8/etc/network/interfaces

./psh "rm /mnt/sda8/etc/udev/rules.d/70-persistent-net.rules"
sleep 2s

#Instalacion del cargador de arranque en los servidores.
./psh "/root/grub.sh"

#Asignacion del nombre del maquina a los servidores.
./hnode.sh

#Reinicio de la utilidad dnsmasq en modo DHCP.
./start-dhcp
#Reinicio de todos los nodos del cluster TFM.
./psh "umount /mnt/sda8"
./psh reboot

echo "*******+*******+*******+*******"
echo "Fin del script configNODE.sh...*******"
echo "*******+*******+*******+*******"
```

Script *ConfigNODE.sh*

- **Instalación del NIS.**

Como etapa final en la instalación del nodo Master, hemos instalado en dicho nodo un servicio de NIS, el cual nos permite administrar los usuarios creados en el sistema.

Para lograr esto hemos instalado el paquete:

```
apt-get install nis
```

Para añadir usuarios en el servidor Master usamos el script useradd.sh visto más adelante en el apartado 4.3 Administración del Sistema.

- **Instalación de Condor y MPI.**

También se han instalado los paquetes de Condor y MPI, tanto en el Master como en los Servidores, para que el Cluster ejecute tareas con fines específicos.

```
apt-get install condor
```

```
apt-get install lam-runtime
```

De forma general, podemos lanzar peticiones al Condor y estas son ejecutadas satisfactoriamente.

He aquí ejemplos de las órdenes que hemos usado al ejecutar tareas en Condor.

- condor_submit: enviamos trabajos a Condor.
- condor_q: controlamos el estado de los trabajos que hemos enviado.
- condor_rm: borrar un trabajo.
- condor_status: obtenemos información del estado de Condor.

4.3. Administración del Sistema.

Ahora vamos a describir las herramientas que nos permiten administrar nuestro sistema. Detallamos los scripts que utilizamos para las tareas de mantenimiento y reparación de los nodos del Cluster.

Se han creado una serie de scripts, los cuales nos permiten realizar las tareas de administración de una manera más efectiva y automatizada. En concreto, a seguidas vamos a describir la funcionalidad de cada una de estas.

- **Instalación del sistema en la partición auxiliar.**

Para instalar el sistema en la partición auxiliar, hemos hecho uso de varios scripts, los cuales definiremos en el curso de este apartado. Utilizaremos los scripts *psh* y *pscp* mencionados anteriormente para realizar operaciones desde el Master hacia los nodos servidores. El script psh envía una orden mediante ssh a todos los servidores del Cluster, y el script pscp se encarga de copiar un fichero desde el Master a los nodos esclavos.

Para realizar la instalación en la partición auxiliar en el nodo Master utilizamos el script *inst-part-mant-master.sh*, el cual tiene la función de preparar convenientemente la segunda partición del sistema, modificar los ficheros `/etc/fstab` de ambas particiones (partición principal y partición auxiliar) y copiar el sistema de la partición principal a la auxiliar.

```
#!/bin/sh

#Instalacion del sistema en la segunda particion.

#Formatear la particion.

mkfs /dev/sda10

#Montar la particion.

mkdir /mnt/sda10
mount /dev/sda10 /mnt/sda10

#Clonar la particion instalada.

cp -ax / /mnt/sda10/

#Configuracion fichero /etc/fstab

rm -r /etc/fstab

echo "/dev/sda8      /         ext4     defaults     0    0" >> /etc/fstab
echo "/dev/sda1      none          swap     auto,defaults    0    0" >> /etc/fstab
echo "/dev/sda10     /mnt/sda10  ext4     defaults     0    0" >> /etc/fstab
echo "/dev/sda9      /nfs          ext4     defaults     0    0" >> /etc/fstab

#Configuracion fichero /mnt/sda10/etc/fstab

rm -r /mnt/sda2/etc/fstab

echo "/dev/sda10     /         ext4     defaults     0    0" >> /mnt/sda2/etc/fstab
echo "/dev/sda1      none          swap     auto,defaults    0    0" >> /mnt/sda2/etc/fstab
echo "/dev/sda8      /mnt/sda8   ext4     defaults     0    0" >> /mnt/sda2/etc/fstab
echo "/dev/sda9      /nfs          ext4     defaults     0    0" >> /mnt/sda2/etc/fstab

#Creacion punto de montaje particion 1 en la particion 2.

mkdir /mnt/sda10/mnt/sda8

#Configuracion GRUB.

rm -r /mnt/sda10/boot/grub
ln -s /mnt/sda8/boot/grub /mnt/sda10/boot/grub


echo "*******+*******+*******+*******"
echo "Configuracion completada satisfactoriamente!!*******"
echo "*******+*******+*******+*******"
echo "FIN DEL SCRIPT!!*******"
```

Script *inst-part-mant-master.sh*

La instalación de la partición auxiliar en los nodos servidores se ha realizado desde el Master empleando el script *inst-part-mant-srvs.sh*, cuya finalidad es un poco similar al de la instalación en el Master, pero con la diferencia de que este se realiza en todos los nodos simultáneamente.

```
#!/bin/sh

#Instalacion del sistema en la segunda particion en los nodos esclavos.

#Formatear la particion.

./psh "mkfs /dev/sda10"

#Montar la particion.

./psh "mkdir /mnt/sda10"
./psh "mount /dev/sda10 /mnt/sda10"

#Clonar la particion instalada.

./psh "cp -ax / /mnt/sda10/"

#Configuracion del fichero /mnt/sda10/etc/fstab.

./pscp /root/fstab2 /mnt/sda10/etc/fstab

#Creacion punto de montaje particion 1 en la particion 2.

./psh "mkdir /mnt/sda10/mnt/sda8"

#Configuracion GRUB.

./psh "rm -r /mnt/sda10/boot/grub"
./psh "ln -s /mnt/sda8/boot/grub /mnt/sda10/boot/grub"

#Configuracion fichero /boot/grub/menu.lst

./pscp /boot/grub/menu.lst /boot/grub/menu.lst

echo "*******+*******+*******+********"
echo "Configuracion completada satisfactoriamente!!*******"
echo "*******+*******+*******+********"
echo "FIN DEL SCRIPT!!*******"
```

Script *inst-part-mant-srvs.sh*

- **Agregar un Nodo Servidor.**

Para agregar un nuevo nodo servidor en nuestro Cluster hemos personalizado dos script para realizar dicha tarea. En concreto, estos scripts son *prepnode.sh* y *addnode.sh*. El primero se encarga de agregar el nuevo nodo a los ficheros `/etc/hosts` y `/etc/ethers` en el nodo Master, así como también de arrancar el servidor PXE para que el nuevo nodo arranque por red (WOL). Cabe destacar que debemos pasarle como parámetros el número del nodo y su dirección MAC sin los dos puntos (:).

La segunda herramienta que empleamos es el script *addnode.sh*. Una vez el nuevo nodo a instalar esté arrancado por red, empleamos esta herramienta, la cual tiene como finalidad la instalación de una réplica del nodo Master en la partición principal del nodo esclavo. A este script le pasamos como parámetro el número de nodo.

```
#!/bin/sh

#Preparacion del nodo Master para agregar un nuevo nodo al cluster TFM.
#
#
#Editamos el fichero /etc/hosts para que incluya el nuevo nodo.
#
echo "192.168.1.20$1        server$1        server$1.cluster" >> /etc/hosts

#Editamos el fichero /etc/ethers para que incluya el nuevo nodo.
#
echo "$2:$3:$4:$5:$6:$7 192.168.1.20$1" >> /etc/ethers

#Arrancamos el servidor en modo DHCP + PXE.
#
./start-dhcp-pxe

echo "*******+*******+*******+********"
echo "Operacion realizada satisfactoriamente!!"
echo "Fin del script prepnode.sh..."
```

Script *prepnode.sh*

```
#!/bin/sh

#Agregar un nuevo nodo servidor al Cluster TFM.

#Particionado del disco de los servidores.
#sfdisk -d /dev/sda > /nfs/srv/root/sda.out
#sleep 1s

#Creacion de la tabla de particiones.
#ssh cluster$1 "sfdisk -f /dev/sda < /root/sda.out"
#sleep 1s

#Formatear la particion de arranque.
ssh server$1 "mkfs /dev/sda8"
sleep 1s

#Preparacion de la swap.
ssh server$1 "mkswap /dev/sda1"
ssh server$1 "swapon /dev/sda1"
sleep 1s

#Clonacion del sistema a los nodos servidores.
ssh server$1 "mkdir /mnt/sda8"
ssh server$1 "mount /dev/sda8 /mnt/sda8"
ssh server$1 "cp -ax / /mnt/sda8"
sleep 1s

#Configuracion adicional de los nodos servidores y del nodo Master.

#Copia del fichero /root/fstab en los nodos servidores.
scp /root/fstab server$1:/mnt/sda8/etc/

#Copia del fichero /root/interfaces en los nodos servidores.
scp /root/interfaces server$1:/mnt/sda8/etc/network/interfaces

ssh server$1 "rm /mnt/sda8/etc/udev/rules.d/70-persistent-net.rules"

#Copia del fichero /etc/hosts en los nodos servidores.
scp /etc/hosts server$1:/mnt/sda8/etc/hosts

#Instalacion del cargador de arranque en los servidores.
ssh server$1 "/root/grub.sh"

#Asignacion de nombre de maquina a los servidores.
ssh server$1 "echo server$1 > /mnt/sda8/etc/hostname"

#Reinicio de la utilidad dnsmasq en modo DHCP.
./start-dhcp

#Reinicio del nodo del Cluster TFM.
ssh server$1 "umount /mnt/sda8"
ssh server$1 reboot

echo "*******+*******+*******+*******"
echo "Configuracion completada satisfactoriamente!!*******"
echo "*******+*******+*******+*******"
echo "Fin del script addnode.sh...*******"
```

Script *addnode.sh*

- **Reparación partición auxiliar en un nodo servidor.**

Esta tarea la realiza el script *rep-part-mant-srvs.sh*, y tiene por finalidad reinstalar la partición auxiliar en un nodo específico, el cual pasamos como parámetro al script. Realiza la misma función la herramienta de instalación en la partición auxiliar en los nodos servidores, pero con la variante de que este repara un nodo en específico.

```
#!/bin/sh

#Instalacion del sistema en la segunda particion en los nodos esclavos.

#Formatear la particion.

ssh server$1 "mkfs /dev/sda10"

#Montar la particion.

ssh server$1 "mkdir /mnt/sda10"
ssh server$1 "mount /dev/sda10 /mnt/sda10"

#Clonar la particion instalada.

ssh server$1 "cp -ax / /mnt/sda10/"

#Configuracion fichero /mnt/sda10/etc/fstab

scp /root/fstab2 server$1/mnt/sda10/etc/fstab

#Creacion punto de montaje particion 1 en la particion 2.

ssh server$1 "mkdir /mnt/sda10/mnt/sda8"

#Configuracion GRUB.

ssh server$1 "rm -r /mnt/sda10/boot/grub"
ssh server$1 "ln -s /mnt/sda8/boot/grub /mnt/sda10/boot/grub"

#Configuracion fichero /boot/grub/menu.lst

scp /boot/grub/menu.lst server$1:/boot/grub/menu.lst

echo "*******+*******+*******+********"
echo "Configuracion completada satisfactoriamente!!*******"
echo "*******+*******+*******+********"
echo "FIN DEL SCRIPT!!********"
```

Script *rep-part-mant-srvs.sh*

- **Creación de un nuevo usuario.**

Para la creación de un nuevo usuario en el sistema, y su posterior actualización en la base de datos del NIS, empleamos un script denominado como *useradd.sh*. Este tiene como función la creación de un usuario pasado como parámetro al script y luego actualizarlo en la base de datos del NIS.

```
#!/bin/sh

#Creacion de un nuevo usuario.

useradd $1 -m

#Asignacion de password.

passwd $1

#Actualizacion de la base de datos NIS.

make -C /var/yp

echo "Usuario" $1 "ha sido creado satisfactoriamente!!!!!!!"
```

Script *useradd.sh*

4.4. Almacenamiento.

Tal y como se ha descrito en el capítulo de Configuración del Cluster, tenemos habilitado un sistema de almacenamiento NAS (Network Attached Storage) el cual provee espacio para el almacenamiento de nuestros datos. El sistema ofrece un RAID 0 con 7,5 TB.

Para configurar este servicio solo basta con modificar el fichero */etc/fstab* de nuestro sistema, y agregar la siguiente línea en todos los nodos.

```
192.168.1.100:/raid0/data/lv0001 /nfs  nfs  nfsvers=3,auto,rw,
wsize=8192,rsize=8192      0     0
```

Nótese que en los scripts de instalación y mantenimiento de los nodos servidores, se copia un fichero */etc/fstab* previamente preparado para ofrecer esta configuración.

4.5. Equilibrado de Carga.

El equilibrado de la carga hace referencia al método por el cual los datos se distribuyen a través de más de un servidor. Casi todas las aplicaciones paralelas o distribuidas se pueden beneficiar del equilibrado de la carga. Normalmente en los entornos Linux, como en casi todos los entornos homogéneos, es controlado por un nodo maestro. Los datos los controla el nodo maestro y se sirven a dos o más máquinas dependiendo del tráfico. Los datos no tienen que distribuirse por igual, por supuesto. Si tenemos un servidor en una gigabit-ethernet, dicho servidor puede obviamente absorber más tráfico que un nodo fast-ethernet.

Una de las ventajas del equilibrado de la carga es que los equilibradores normalmente reaccionan ante los fallos redireccionando el tráfico desde el nodo de bajada y distribuyendo los datos por los nodos restantes. La mayor desventaja del equilibrado de la carga es que los datos tienen que mantenerse consistentes y disponibles para todos los servidores, aunque se podría utilizar un método como *rsync* para mantener la integridad de aquellos.

En este apartado se describe el procedimiento que se ha seguido para realizar la configuración de LVS-NAT en el Cluster. LVS-NAT es el modo más simple de configurar LVS. La funcionalidad de este consiste en lo siguiente: los paquetes de los clientes llegan al director donde la dirección VIP (Virtual IP) se cambia por la dirección RIP (Real IP) de uno de los servidores reales. A su vez, las respuestas de los servidores reales pasan por el director que sustituye su dirección RIP por la dirección VIP.

Hemos usado como directores a los nodos **Master** y **Standby**, y el resto de los servidores (server2, server3, server4, server5 y server6) como servidores reales. Esto hecho así, debido a que los nodos de cómputo son los encargados de ofrecer el servicio web (HTTP) y el servicio de máquinas virtuales, mientras que los directores son los encargados de proporcionar el Equilibrado de Carga y la Alta Disponibilidad (HA).

- **Instalación de LVS en el director.**

En primer lugar, se ha realizado la instalación del paquete *ipvsadm*.

```
apt-get install ipvsadm
```

Se ha configurado la interfaz de red eth1 como interfaz multicast IPVS. Esto se logra ejecutando el siguiente mandato:

```
dpkg-reconfigure ipvsadm
```

Luego verificamos que se encuentra habilitado el ip_forwarding, esto es, que el fichero `/proc/sys/net/ipv4/ip_forward` tiene un uno (1) como contenido.

Luego de esto, el segundo paso realizado es configurar la (VIP) y las RIP (Real IP) en el fichero `/etc/ipvsadm.rules`. Cuyo contenido ha quedado de la siguiente manera:

```
-A -t 158.42.181.221:80 -s rr
-a -t 158.42.181.221:80 -r 192.168.1.202:80 -m
-a -t 158.42.181.221:80 -r 192.168.1.203:80 -m
-a -t 158.42.181.221:80 -r 192.168.1.204:80 -m
-a -t 158.42.181.221:80 -r 192.168.1.205:80 -m
-a -t 158.42.181.221:80 -r 192.168.1.206:80 -m
```

La primera línea se corresponde a la adición de un servicio virtual para la VIP 158.42.181.221 al puerto 80 y el método de planificación es el de round robin (-s rr). rr es un **Scheduling-Method** (Método de Planificación) empleado por ipvsadm, que tiene como fin distribuir los trabajos en partes iguales entre los servidores disponibles.

Las siguientes cinco líneas se corresponden con la configuración de las RIP del Cluster. La función de esta configuración es redirigir los paquetes de la VIP a la dirección de los servidores reales (RIP) dentro del Cluster, y especificamos que el método de retransmisión de paquetes es NAT (opción –m, que significa masquerading).

- **Instalación de Apache y PHP.**

En los servidores esclavos lo único que se ha hecho es instalar y configurar los paquetes *apache2* y *php5* para que puedan ofrecer el servicio web y el sistema de máquinas virtuales. La instalación y configuración del servicio de Máquinas Virtuales lo abordaremos en el apartado 4.7 de este escrito.

Para instalar Apache y Php en los servidores esclavos, los cuales como ya hemos mencionado son los que ofrecerán el servicio web y de máquinas virtuales, hemos empleado el script psh para realizar la instalación desde el nodo Master a todos los servidores.

```
psh apt-get install apache2
```

```
psh apt-get install php5
```

Luego debemos configurar en cada servidor el fichero: `/etc/apache2/sites-enabled/000-default` y agregar la siguiente línea:

```
DocumentRoot      /websrv/www/
```

El directorio */websrv* se encuentra en nuestro nodo Master, y es compartido con los demás nodos del Cluster mediante NFS.

Tenemos disponible el fichero de configuración de apache en el nodo Master, el cual modificamos y enviamos a los nodos servidores empleando el script pscp.

4.6. Alta Disponibilidad.

Los ordenadores tienen una molesta tendencia a fallar cuando menos lo esperamos. Es raro no encontrarse con un administrador de sistemas que no haya recibido una llamada en mitad de la noche con la mala noticia de que un sistema crítico ha caído.

Los conceptos de alta disponibilidad y clusters tolerantes a fallos tienden a ir de la mano. Si un sistema va a conseguir altos tiempos de funcionamiento, harán falta más subsistemas de redundancia para mantenerlo operativo (incluso la adición de servidores en una configuración Cluster). El punto de partida para los clusters de alta disponibilidad supone que las aplicaciones son de tal importancia que hay que tomar medidas extra para asegurarse que está disponible.

La solución adoptada en este proyecto para proveer a nuestro Cluster de Alta Disponibilidad ha consistido en agregar un nodo adicional al Cluster. Este nodo posee las mismas características del nodo Master. A dicho nodo le hemos llamado Standby, y es una réplica del Master, lo cual nos asegura que es un suplente genuino de este. En la siguiente tabla se muestra la distribución de nuestros nodos Directores y nuestros Servidores Reales con sus respectivas RIP (Real IP) y la VIP (Virtual IP) que funciona como pasarela al exterior para los Servidores Reales.

	NOMBRE	DIRECCION IP			
		RIP Interna	VIP Interna	RIP Externa	VIP Externa
DIRECTORES	Master (server1)	192.168.1.201	192.168.1.10	158.42.181.221	158.42.181.218
	Standby (server7)	192.168.1.207		158.42.181.220	
SERVIDORES REALES	server2	192.168.1.202	-	-	-
	server3	192.168.1.203	-	-	-
	server4	192.168.1.204	-	-	-
	server5	192.168.1.205	-	-	-
	server6	192.168.1.206	-	-	-

Para conseguir la configuración correcta de Alta Disponibilidad con Equilibrado de Carga, instalamos los paquetes *corosync, pacemaker* y *ldirectord* en ambos directores (Master y Standby). Una vez instalados, lo primero que hacemos es activar el corosync editando el fichero `/etc/default/corosync`. Y reiniciamos el servicio: */etc/init.d/corosync start.*

- **Configuración de ldirectord.**

Para configurar el servicio ***ldirectord*** deshabilitamos los scripts de inicio de ldirectord e ipvsadm, ya que ahora será pacemaker quien se encargue de lanzar ldirectord en la máquina que esté activa, y a su vez, ldirectord lanzará ipvsadm con los parámetros adecuados.

Para configurar ldirectord editamos el fichero: `/etc/ha.d/ldirectord`. En este fichero debemos especificar la VIP y las RIP de nuestros servidores. El fichero ha quedado con el siguiente contenido:

```
# Global Directives
checktimeout=10
checkinterval=2
#fallback=127.0.0.1:80
autoreload=no
#logfile="/var/log/ldirectord.log"
logfile="local0"
#emailalert="admin@x.y.z"
#emailalertfreq=3600
#emailalertstatus=all
quiescent=yes

# Sample for an http virtual service
virtual=158.42.181.218:80
        real=192.168.1.202:80 masq
        real=192.168.1.203:80 masq
        real=192.168.1.204:80 masq
        real=192.168.1.205:80 masq
        real=192.168.1.206:80 masq
        fallback=127.0.0.1:80 gate
        service=http
        request="index.html"
        receive="Test Page"
        virtualhost=some.domain.com.au
        scheduler=rr
        #persistent=600
        #netmask=255.255.255.255
        protocol=tcp
        checktype=connect
        checkport=80
        request="index.html"
        receive="Test Page"
        virtualhost=www.x.y.z
```

- **Configuración de pacemaker.**

Ahora definiremos los recursos del Cluster que deben ser gestionados por *pacemaker*. En concreto, son tres:

- La IP Virtual en la red externa: debido a que el servicio de distribución de carga puede migrar entre el Master (158.42.181.221) y Standby (158.42.181.220), hemos definido una IP Virtual (158.42.181.218) la cual será un alias de la máquina que, en cada momento esté dando servicio. Esta será la dirección “pública” de nuestro Cluster y a la que accederán los clientes.
- La IP Virtual en la red interna: ya que el servicio de distribución de carga puede migrar entre el Master (192.168.1.201) y Standby (192.168.1.207), debemos definir una IP Virtual (192.168.1.210), la cual será un alias en la red interna de la máquina que, en cada momento, se encuentre proporcionando el servicio. Esto es necesario para el caso en que uno de los distribuidores de carga falle. El distribuidor superviviente

deberá hacer de "gateway" de los servidores reales. Por tanto es necesario que los servidores reales tengan como gateway la IP: 192.168.1.210.

- El propio servicio de distribución de carga ldirectord: este es el encargado de lanzar IPVS (LVS), configurándolo para que distribuya la carga sobre los servidores reales que en cada momento estén activos.

La lograr todo esto, debemos editar el fichero correspondiente lanzando la siguiente orden: `crm configure edit`. A continuación se muestra el contenido de este fichero:

```
node server1
node standby
primitive ip1 ocf:heartbeat:IPaddr2 \
        params ip="158.42.181.218" nic="eth1" cidr_netmask="24" broadcast="158.42.181.255"
primitive ip2 ocf:heartbeat:IPaddr2 \
        params ip="192.168.1.210" nic="eth0" cidr_netmask="24" broadcast="192.168.1.255"
primitive ldirectord1 ocf:heartbeat:ldirectord \
        params configfile="/etc/ha.d/ldirectord.cf" \
        op monitor interval="15s" timeout="20s" \
        meta migration-threshold="10" target-role="Started"
group group1 ip1 ip2 ldirectord1
order ip_before_lvs inf: ip1:start ip2:start ldirectord1:start
property $id="cib-bootstrap-options" \
        dc-version="1.1.6-9971ebba4494012a93c03b40a2c58ec0eb60f50c" \
        cluster-infrastructure="openais" \
        stonith-enabled="false" \
        no-quorum-policy="ignore" \
        expected-quorum-votes="2"
~
```

Por último, para configurar el nuevo gateway de los servidores reales hemos editado el fichero `/etc/dnsmasq.conf` y añadimos la línea: `dhcp-option=3,192.168.1.210`. Y reiniciamos el servicio *dnsmasq*: `service dnsmasq restart`.

4.7. Sistema de Máquinas Virtuales.

Una vez implementado en nuestro Cluster el servicio Web de Alta Disponibilidad con Equilibrado de Carga, lo siguiente que hemos incorporado al Cluster es un Sistema de Máquinas Virtuales. Este sistema nos permite acceder a las máquinas virtuales del Cluster desde un ordenador remoto, o de manera local. A continuación describimos el procedimiento que se ha seguido para lograr la implementación del servicio de virtualización.

En primer lugar, hemos instalado el software de virtualización **Oracle VM VirtualBox** en todos los nodos de cómputo que conforman el Cluster. También, se ha instalado convenientemente el paquete de Virtual Box conocido como **Extension Pack**, el cual nos brinda la posibilidad de hacer que Virtual Box sea compatible con diversas arquitecturas de ordenadores, y con múltiples dispositivos de entrada y salida, así como también el ***VirtualBox RDP*** (Remote Desktop Protocol) que nos permite acceder remotamente a las máquinas virtuales.

Luego de esto, se ha instalado el paquete **PhpVirtualBox**, tanto en los nodos directores como en los servidores reales, para el ofrecimiento del servicio de virtualización remotamente. PhpVirtualBox es una implementación Open Source programado en Ajax/PHP que nos permite administrar nuestras máquinas virtuales de Virtual Box mediante una interfaz web, ya sea accediendo de forma local o remotamente.

Para lograr la configuración de PhpVirtualBox, primero debemos agregar un usuario de nuestro sistema al grupo *vboxusers*, por ejemplo, hemos creado el usuario vbox y lo agregamos al grupo vboxusers. Este usuario es el que ejecutará las máquinas virtuales en el servidor.

Una vez realizado esto, debemos modificar el fichero */etc/default/virtualbox* en nuestro Master y agregamos las siguientes líneas:

```
VBOXWEB_USER=vbox
VBOXWEB_HOST=192.168.1.201
```

Tambien debemos hecer lo mismo en los servidores, pero cambiando la direccion IP por la direccion correspondiente.

Una vez hecho esto arrancamos el servicio de máquinas virtuales:

```
/etc/init.d/vboxweb-service restart
```

Luego, se ha modificado apropiadamente el fichero */websrv/www/phpvirtualbox/config.php* en el cual debemos agregar el usuario creado anteriormente y la contraseña en las líneas *username* y *password*.

También en este fichero hemos agregado un "Array" de servidores que contiene todos los servidores de nuestro cluster.

En la siguiente figura mostramos el contenido del fichero:

```
<?php
/**
 * phpVirtualBox example configuration.
 * @version $Id: config.php-example 452 2012-10-17 12:22:12Z imooreyahoo@gmail.com $
 *
 * rename to config.php and edit as needed.
 *
 */
class phpVBoxConfig {

/* Username / Password for system user that runs VirtualBox */
var $username = 'vbox';
var $password = 'vbox';

/* SOAP URL of vboxwebsrv (not phpVirtualBox's URL) */
var $location = 'http://127.0.0.1:18083/';

/* Default language. See languages folder for more language options.
 * Can also be changed in File -> Preferences -> Language in
 * phpVirtualBox.
 */
var $language = 'en';

/* Set the standard VRDE Port Number / Range, e.g. 1010-1020 or 1027 */
var $vrdeports = '9000-9100';
```

/websrv/www/phpvirtualbox/config.php.

```
var $servers = array(
        array(
                'name' => 'Server1',
                'username' => 'vbox',
                'password' => 'vbox',
                'location' => 'http://192.168.1.201:18083/',
                'authMaster' => true // Use this server for authentication
        ),
        array(
                'name' => 'Server2',
                'username' => 'vbox',
                'password' => 'vbox',
                'location' => 'http://192.168.1.202:18083/',
        ),
        array(
                'name' => 'Server3',
                'username' => 'vbox',
                'password' => 'vbox',
                'location' => 'http://192.168.1.203:18083/',
        ),
        array(
                'name' => 'Server4',
                'username' => 'vbox',
                'password' => 'vbox',
                'location' => 'http://192.168.1.204:18083/',
        ),
        array(
                'name' => 'Server5',
                'username' => 'vbox',
                'password' => 'vbox',
                'location' => 'http://192.168.1.205:18083/',
        ),
        array(
                'name' => 'Server6',
                'username' => 'vbox',
                'password' => 'vbox',
                'location' => 'http://192.168.1.206:18083/',
        ),
```

Fichero /websrv/www/phpvirtualbox/config.php

5. PRUEBAS

En este capítulo describimos el conjunto de pruebas realizadas sobre el Cluster implementado. Estas pruebas se han ejecutado para, en cierta medida, evaluar el desempeño de los servicios prestados por el Cluster.

Se han hecho pruebas tanto para el servicio web, como para el servicio de máquinas virtuales. Para el servidor web hemos evaluado el funcionamiento del Reparto de Carga y la Alta Disponibilidad. Mientras que para el servicio de máquinas virtuales se ha constatado que los nodos son capaces de arrancar las máquinas y brindarlas a los usuarios remotos.

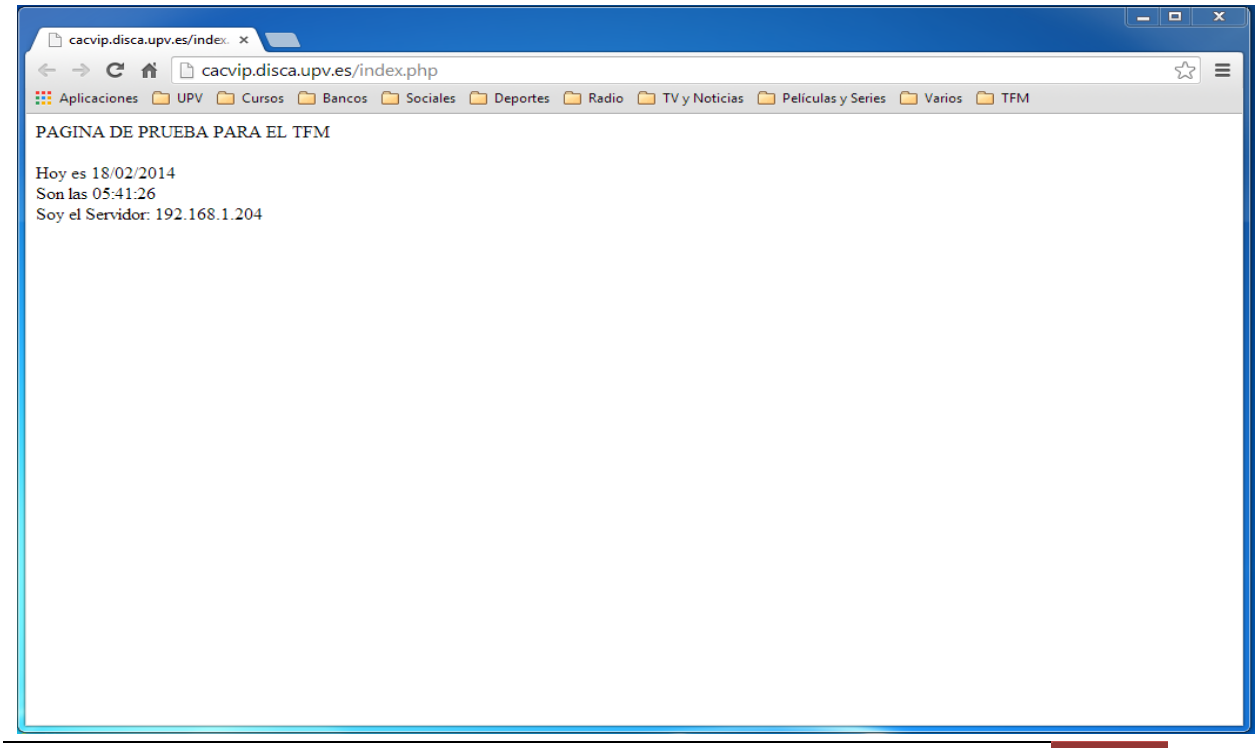
5.1. Servidor Web.

En este apartado detallamos los experimentos realizados al servicio web que proporciona el Cluster implementado. Hemos hecho uso de varias herramientas y utilidades que nos han permitido evaluar dicho servicio para comprobar sus prestaciones. Estas herramientas las iremos definiendo en el trascurso del apartado.

5.1.1. Reparto de Carga

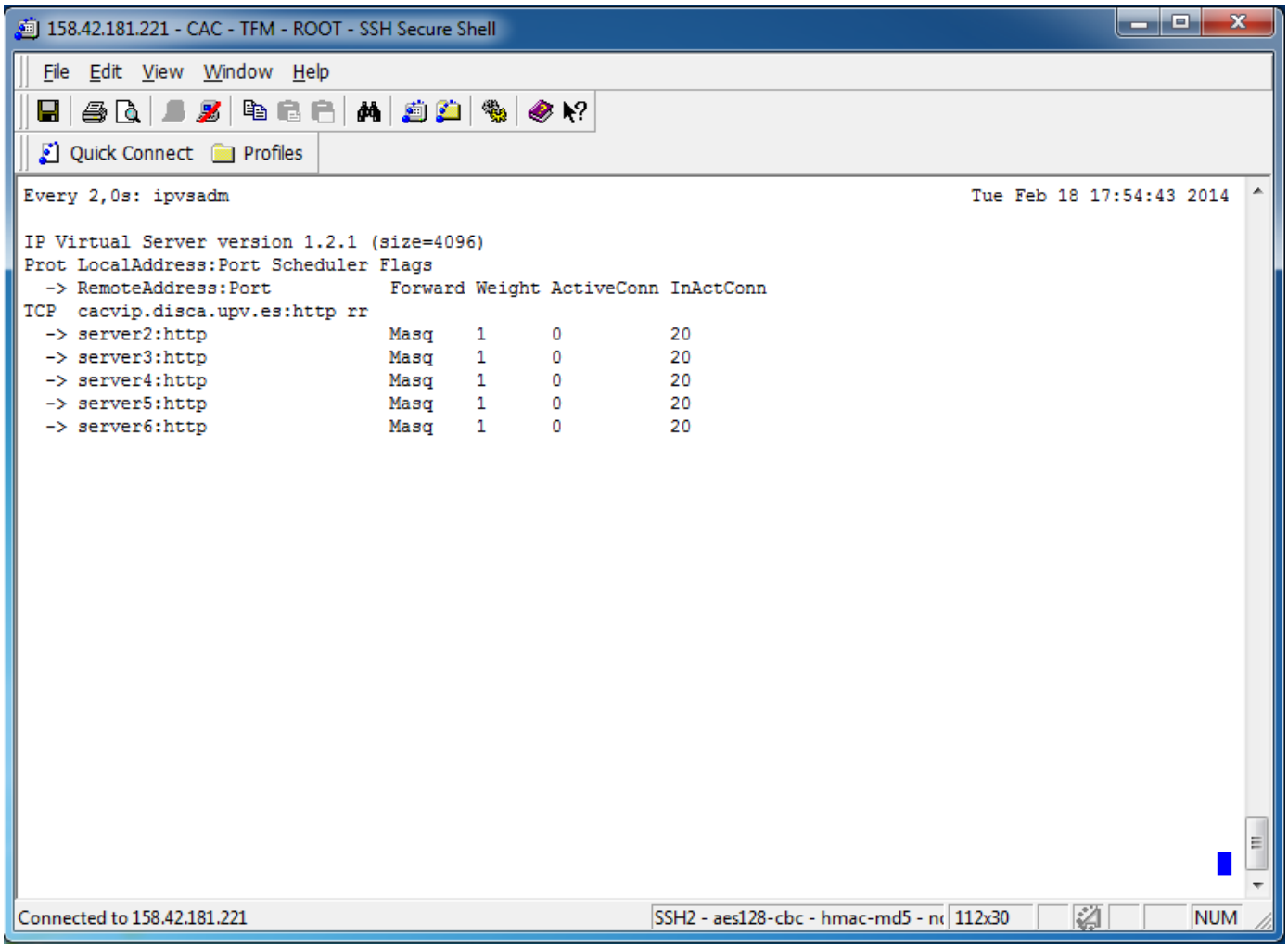
Para evaluar el reparto de carga en el Cluster hemos efectuado los siguientes experimentos:

1. Como no sabemos cuál nodo nos está brindando la página web en cada momento, hemos creado una sencilla, pero útil, pagina web en Php, que hemos llamado index.php, la cual nos muestra la dirección IP del servidor que en ese instante nos proporciona el servicio. Para visualizar esta página solo basta con acceder a la siguiente URL: *http://cacvip.disca.upv.es/index.php*. En la siguiente figura podemos observar el contenido de dicha página.



Si realizamos varias peticiones sobre la URL anterior, veremos que cambia el servidor que ofrece la página.

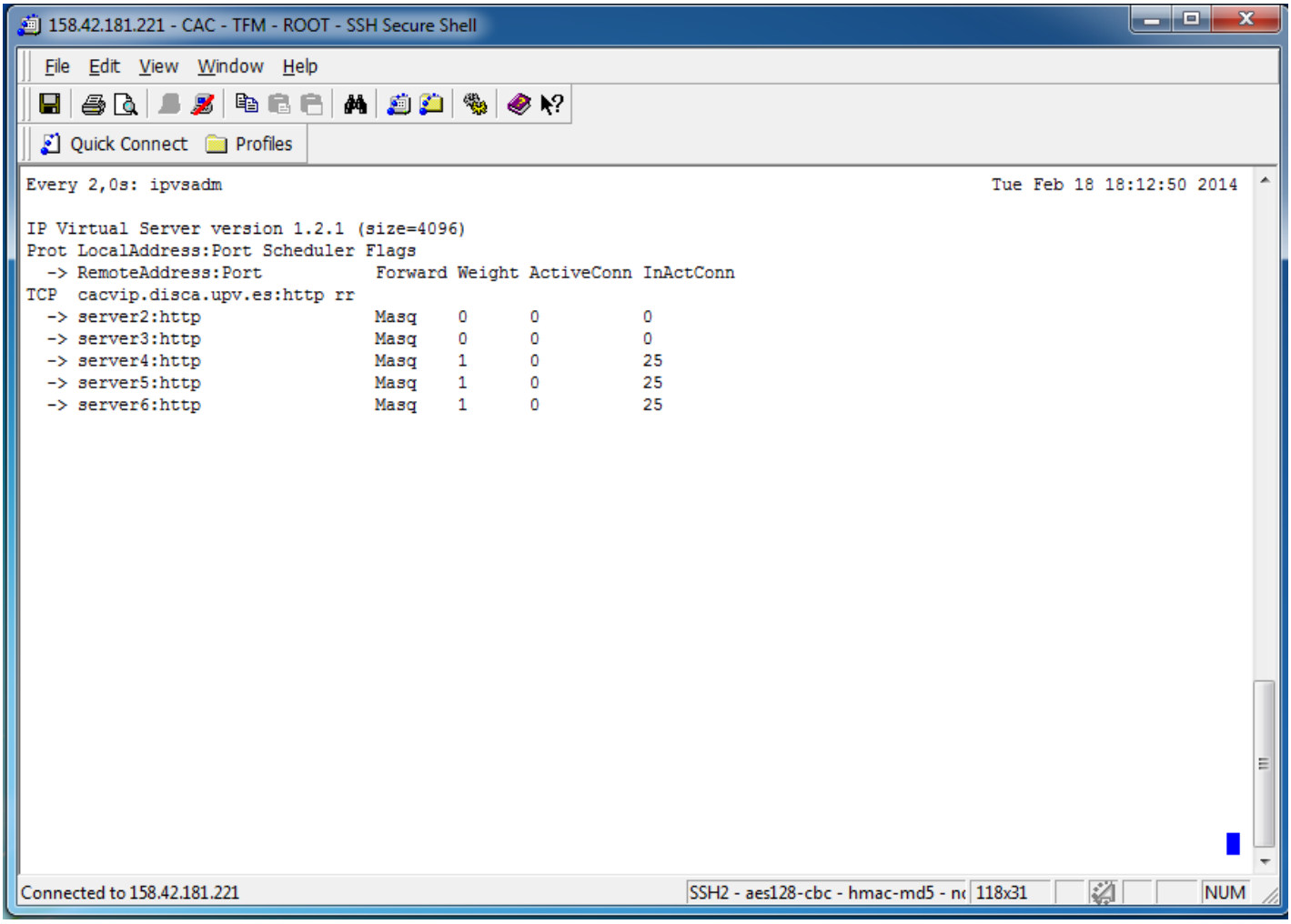
2. Empleando la orden `watch ipvsadm` en el nodo Master, comprobamos cómo se va actualizando la tabla de conexiones del director. Mediante esta utilidad podemos observar cómo se asignan las peticiones a los servidores reales. Para realizar esta prueba hemos lanzado peticiones hacia el Cluster empleando la URL mencionada anteriormente. En la figura siguiente observamos la carga de los servidores reales.



Utilidad watch ipvsadm (a)

En la figura ***Utilidad watch ipvsadm (a)*** podemos observar que estamos realizando peticiones sobre nuestra dirección VIP. En este experimento tenemos todos los servidores de cómputo en funcionamiento (server2 al server6). En la columna *Weight* apreciamos que todos tienen asignados el mismo peso, es decir, que las peticiones son repartidas equitativamente. Por último, en la columna *InActConn* observamos las peticiones que atienden cada uno simultáneamente.

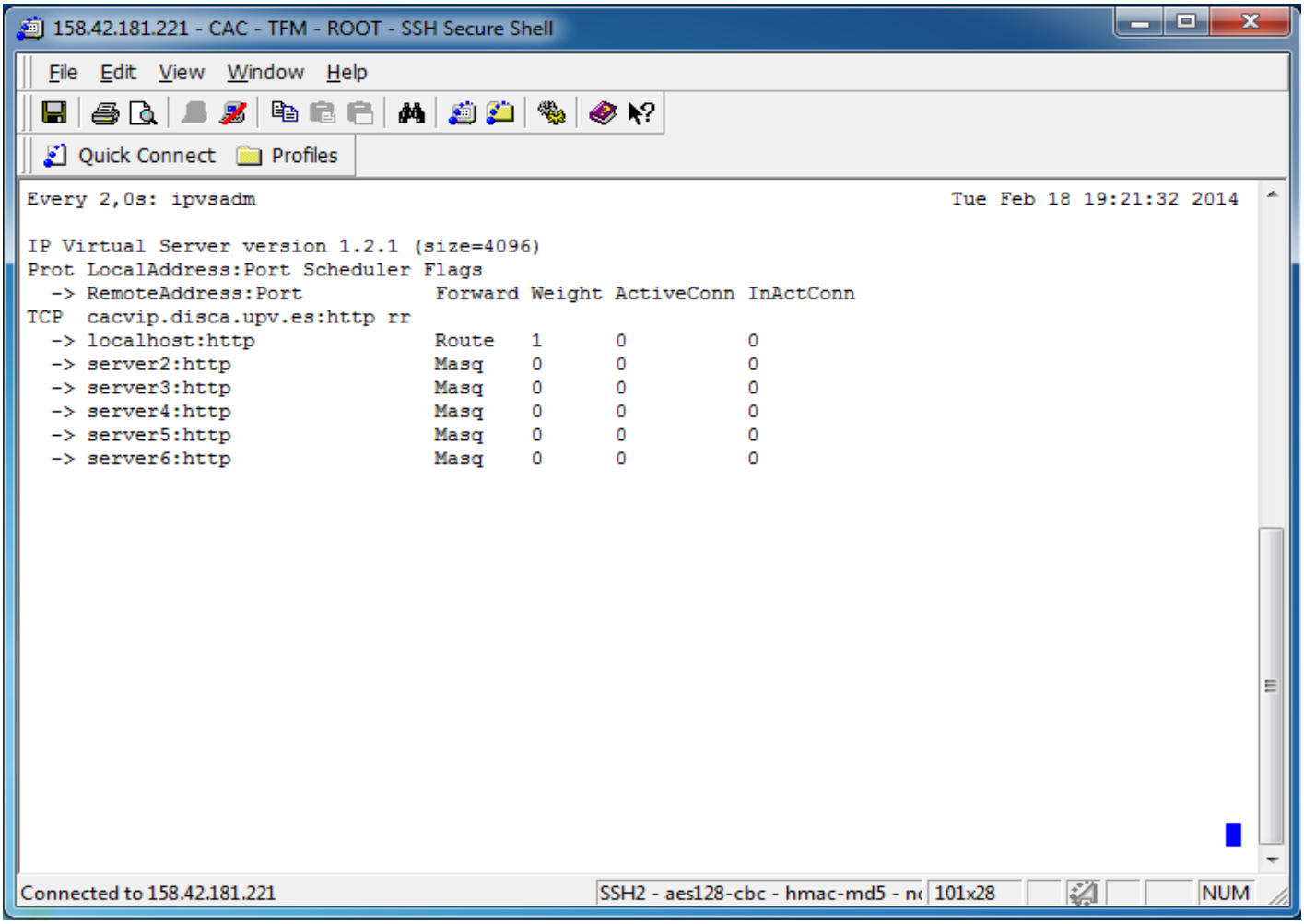
3. El tercer experimento de reparto de carga ha consistido en hacer fallar uno o más nodos para comprobar que el director que se encuentra al mando del servicio en este momento (Master o Standby) tiene la capacidad de repartir la carga sobre los nodos disponibles. En concreto, hemos hecho fallar los servidores: server2 y server3. Nuevamente lanzamos peticiones a la URL de nuestro Cluster y volvemos a ejecutar la orden watch ipvsadm y a continuación, en la siguiente figura se muestra el resultado que nos ofrece la utilidad.



Utilidad watch ipvsadm (b)

La figura ***Utilidad watch ipvsadm (b)*** es un poco similar a la presentada anteriormente. La diferencia es que en esta podemos apreciar que el server2 y el server3 no tienen ningún peso asignado debido a que han fallado. Al realizar el experimento sobre el Cluster observamos que el director reparte la carga entre los tres servidores que tiene disponibles (server4 al server6).

4. En esta cuarta y última prueba que hemos realizado sobre el servicio de Reparto de Carga hemos hecho fallar todos los nodos de cómputo del Cluster (nodos esclavos) para comprobar que el director (Master o Standby) asume el servicio. Lanzamos peticiones a la URL del Cluster y observamos como el director se hace cargo del servicio. En la siguiente figura apreciamos la utilidad watch ipvsadm.



Utilidad watch ipvsadm (c)

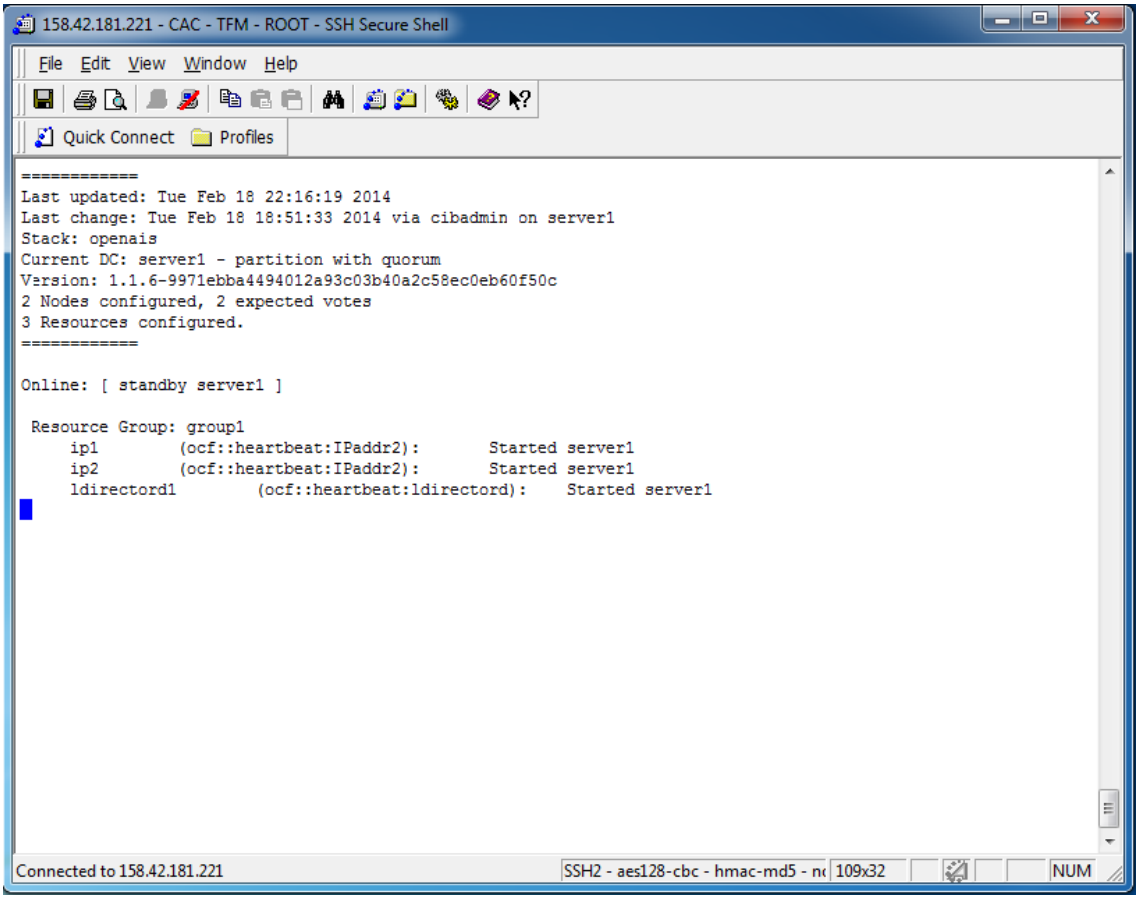
En la figura ***Utilidad watch ipvsadm (c)*** apreciamos que todos los servidores reales se encuentran fuera de línea, y por tanto el servicio recae sobre el localhost. Esto se debe a que el director no tiene ningún servidor real disponible sobre el cual repartir la carga.

5.1.2. Alta Disponibilidad

Para la evaluación de la Alta disponibilidad hemos empleado la orden `crm_mon`. Esta utilidad nos indica cuál de los directores se encuentra disponible en un momento dado. A continuación mostraremos las pruebas que hemos realizado.

- **Master y Standby on-line.**

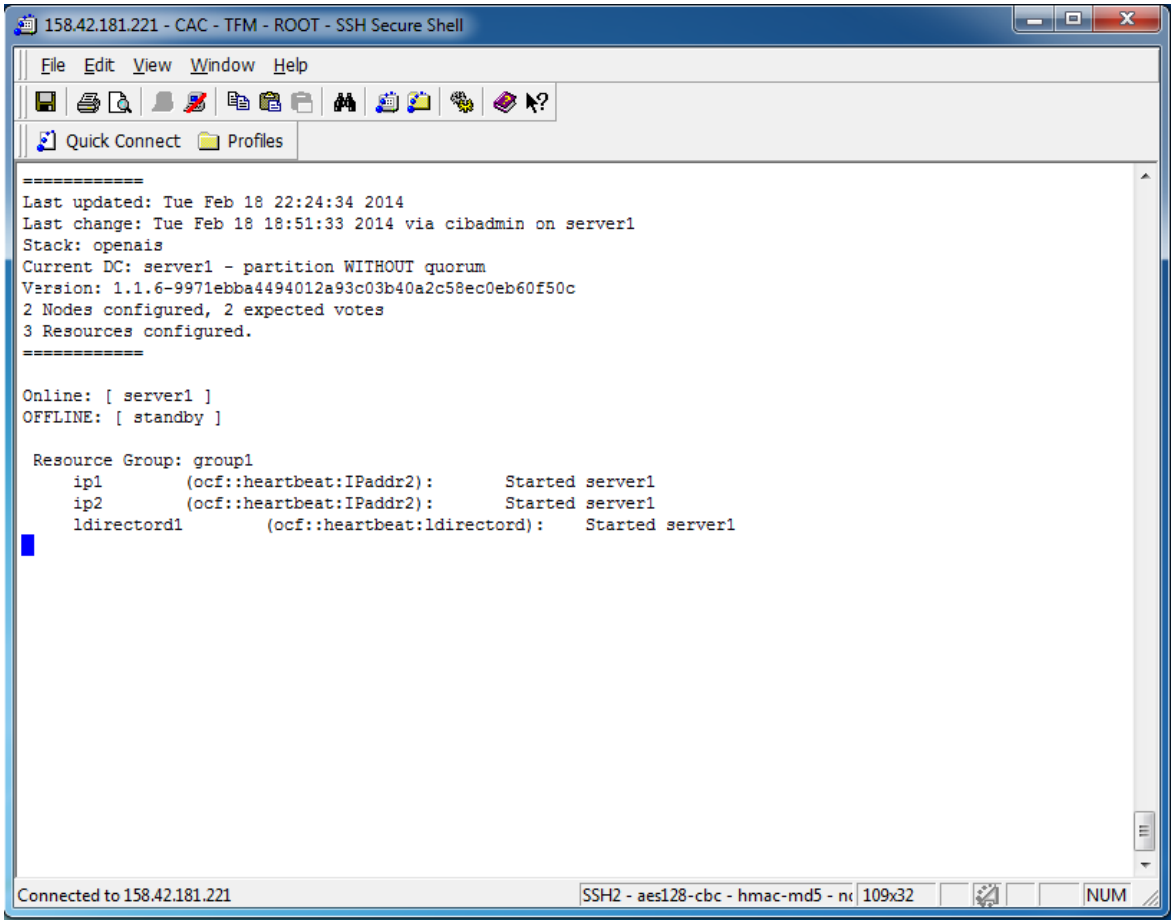
La siguiente figura muestra que tanto ambos directores (Master y Standby) se encuentran en línea.



Utilidad crm_mon Master y Standby (a)

- **Master on-line.**

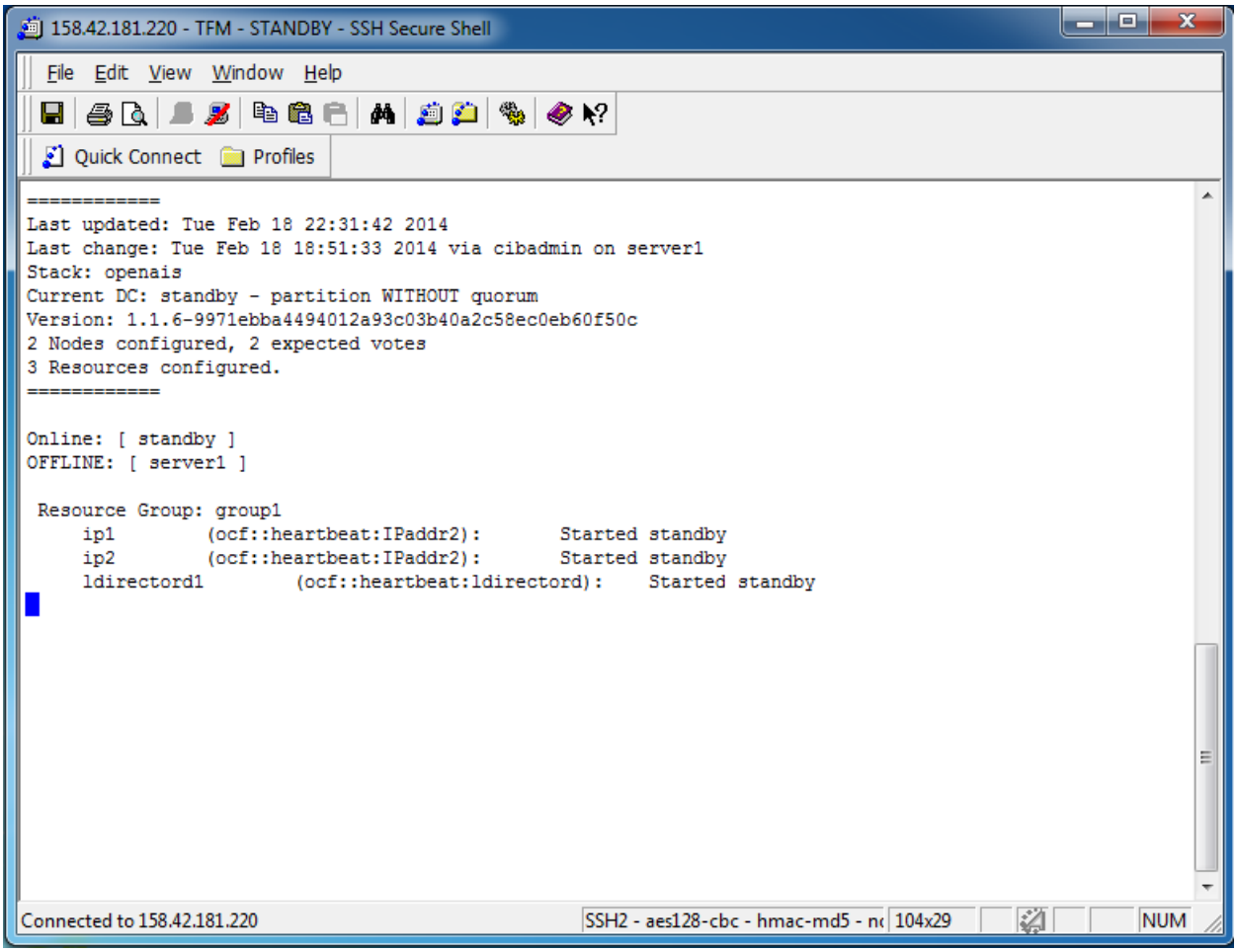
En esta prueba hacemos fallar el nodo Standby. Ejecutando la orden `crm_mon` sobre el Master podemos apreciar que este se encuentra en línea, mientras que Standby está fuera de servicio. Es decir, aun fallando uno de los directores, el servicio web y de máquinas virtuales continua disponible.



Utilidad crm_mon Master (b)

- **Standby on-line.**

Esta prueba es similar a la realizada anteriormente. En este caso hicimos fallar al Nodo Master y comprobamos que el Nodo Standby se ha hecho cargo, proporcionando el servicio web y de máquinas virtuales. Al lanzar desde Standby la orden `crm_mon` podemos apreciar que el Master se encuentra fuera de línea. La figura siguiente muestra la utilidad `crm_mon` ejecutándose desde Standby.



Utilidad crm_mon Master (c)

5.1.3. Evaluación Del Servidor Web.

Para evaluar el reparto de carga que realizan los directores sobre los nodos que proporcionan el servicio web, hemos empleado la herramienta ApacheBench (ab). Ab es una sencilla herramienta para testear servidores web, la cual nos permite medir el rendimiento del servidor. ApacheBench nos permite realizar *"n"* peticiones distribuidas en *"x"* hilos simultáneamente. Esto resulta bastante ventajoso en el sentido de que podemos enviar peticiones a nuestro Cluster para medir su tiempo de respuesta.

Se han realizado pruebas usando dos tipos diferentes de páginas web. En concreto, para una primera evaluación hemos utilizado una página web escrita en HTML y luego se han realizado pruebas empleando tres páginas escritas en PHP. A continuación describimos en que han consistido ambas pruebas.

- **Pruebas del Servidor Web con una Página HTML.**

Para las pruebas con una página escrita en HTML hemos empleado una web modelo denominada **LANC 2011**, la cual es una web de la "Sexta Conferencia de Redes en Latinoamérica 2011" (6th Latin America Networking Conference 2011).

Se ha empleado el ApacheBench para lanzar peticiones desde un ordenador remoto con los siguientes parámetros:

```
ab -n 1000 -c 10 http://cacvip.disca.upv.es/
```

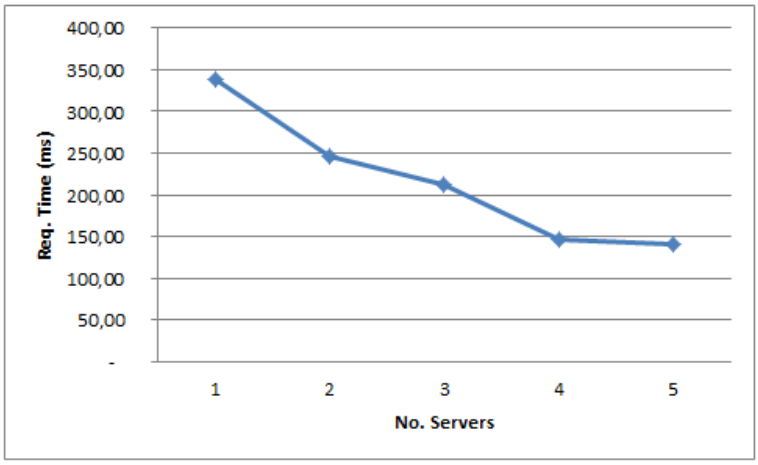
Donde el parámetro 1000 indica la cantidad de peticiones lanzadas, y el parámetro 10 significa la cantidad de clientes o hilos. La URL corresponde a la página web LANC 2011.

En este mismo orden, hemos realizado dos pruebas diferentes empleando uno de los distribuidores de carga a la vez. Es decir, la primera prueba la hemos realizado utilizando al Nodo Master como repartidor de carga y la segunda prueba la hemos hecho empleando al Nodo Standby como distribuidor.

- **Prueba con el Nodo Master.**

1.000,00	Peticiones
10,00	Clientes

Servers	Request Time (ms)
1	338,00
2	246,00
3	212,00
4	146,00
5	141,00



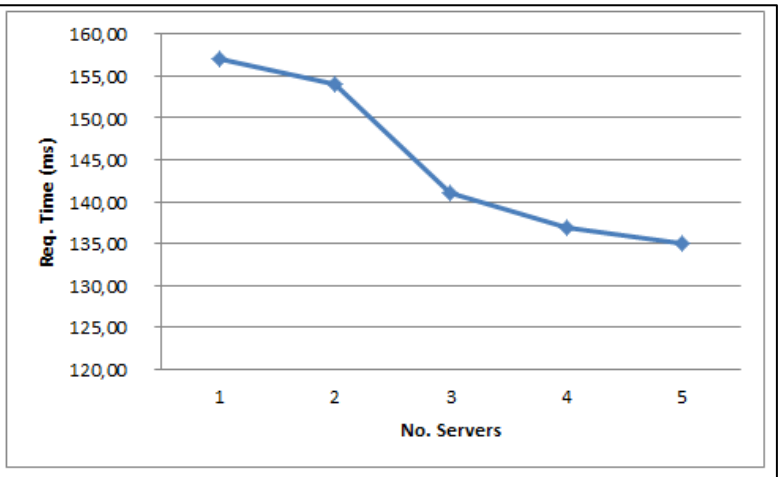
Como se puede observar en la gráfica anterior, se ha realizado un estudio midiendo el tiempo de respuesta del Cluster (ms) empleando al Nodo Master como repartidor de carga. Observamos que a medida que decrece el número de servidores que proveen el servicio, aumenta el tiempo de respuesta.

Este resultado nos lleva a la conclusión de que el repartidor de carga esta realizando una repartición equitativa de las peticiones que recibe.

- **Prueba con el Nodo Standby.**

1.000,00	Peticiones
10,00	Clientes

Servers	Request Time (ms)
1	157,00
2	154,00
3	141,00
4	137,00
5	135,00



En esta prueba hemos utilizado solamente al Nodo Standby como distribuidor de la carga. Observando la gráfica anterior apreciamos un comportamiento un poco similar a la prueba anterior, donde a medida que disminuimos la cantidad de servidores, aumenta el tiempo de respuesta.

Aunque en este estudio los tiempos de respuesta son más pequeños que el caso anterior, esto se puede dar por condiciones del tráfico en la red.

- **Pruebas del Servidor Web con una Página Dinámica (PHP).**

En este apartado hemos realizado varias pruebas sobre una página dinámica escrita en php para validar el tiempo de respuesta de nuestro servidor web. En este experimento empleamos la herramienta ApacheBench (ab) descrita en el apartado anterior. Para realizar dichas pruebas, empleamos un programa para el cómputo de Pi, el cual consiste en calcular dicha constante mediante una integral.

En la figura pi.php podemos observar el algoritmo utilizado en la página dinámica para el cálculo de Pi.

```php
<html>
<body>
<?php
$start=microtime(true);

$area=0.0;

$n=$_GET["n"];
// $n=100000000;

for ($i=0; $i<$n; $i++)
    {
    $x=($i+0.5)/$n;
    $area=$area+4.0/(1.0+$x*$x);
    }
$result=$area/$n;

$end=microtime(true);
$exectime=$end-$start;

echo "<br>Calculo de PI<br><br>";
printf ("La cte. PI con n= %d es igual a %f<br>", $n, $result);
printf ("Tiempo de ejecucion= %.5f segundos<br>",$exectime);
printf ("<br>El servidor es %s<br>", $_SERVER['SERVER_ADDR']);
?>
</body>
</html>
```

pi.php

Como se puede apreciar, para la realización del cálculo se toma un número de iteraciones "n", y se calcula el Pi mediante una integral. Esta página nos da como resultado el valor de la constante PI, el tiempo de ejecución en segundos y el servidor que ha ejecutado la petición.

En nuestro caso, como empleamos LVS para el reparto de la carga, el tiempo de ejecución se ve afectado al variar el número de servidores que ofrece el servicio.

Para efectuar las pruebas hemos utilizado los siguientes parámetros:

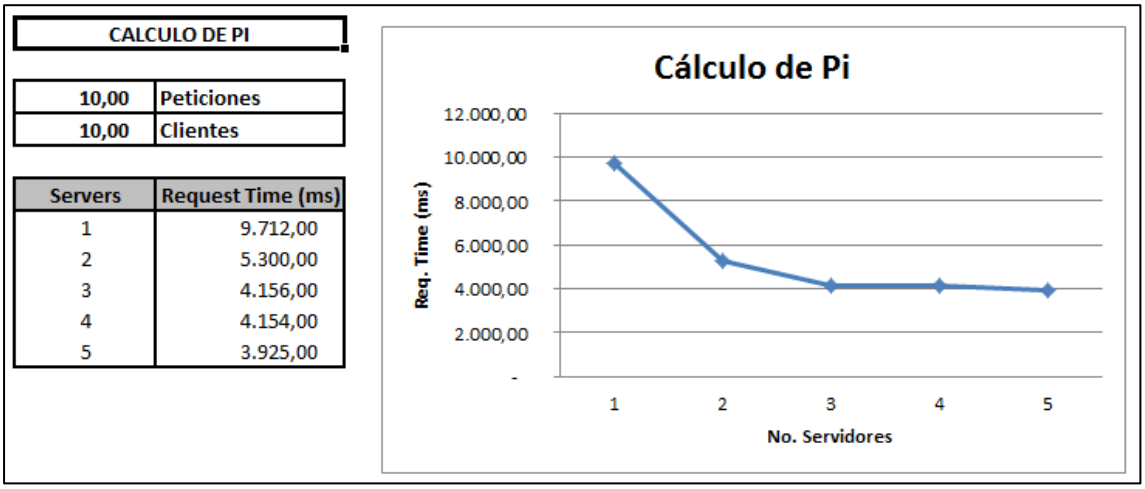
- **Número de iteraciones:** 1000000
- **Número de peticiones:** 10
- **Número de clientes:**10

En concreto, para realizar la prueba ejecutamos la siguiente orden desde un ordenador remoto:

```
ab -n 10 -c 10 http://cacvip.disca.upv.es/pi.php?n=1000000
```

De esta manera lanzamos a nuestro servidor web 10 peticiones repartidas en 10 clientes o hilos simultáneamente, y el número de iteraciones para el cálculo de Pi = 1000000 (n=1000000).

En la figura cálculo de pi podemos observar el comportamiento del tiempo de respuesta en nuestro servidor web. Apreciamos que a medida que aumentamos el número de servidores brindando el servicio, disminuye el tiempo de respuesta del mismo.



CALCULO DE PI

10,00	Peticiones
10,00	Clientes

Servers	Request Time (ms)
1	9.712,00
2	5.300,00
3	4.156,00
4	4.154,00
5	3.925,00

Cálculo de pi

5.2. Sistema de Máquinas Virtuales.

En este apartado vamos a validar que tenemos acceso a nuestro sistema de máquinas virtuales, las cuales podemos ejecutar perfectamente desde un ordenador remoto.

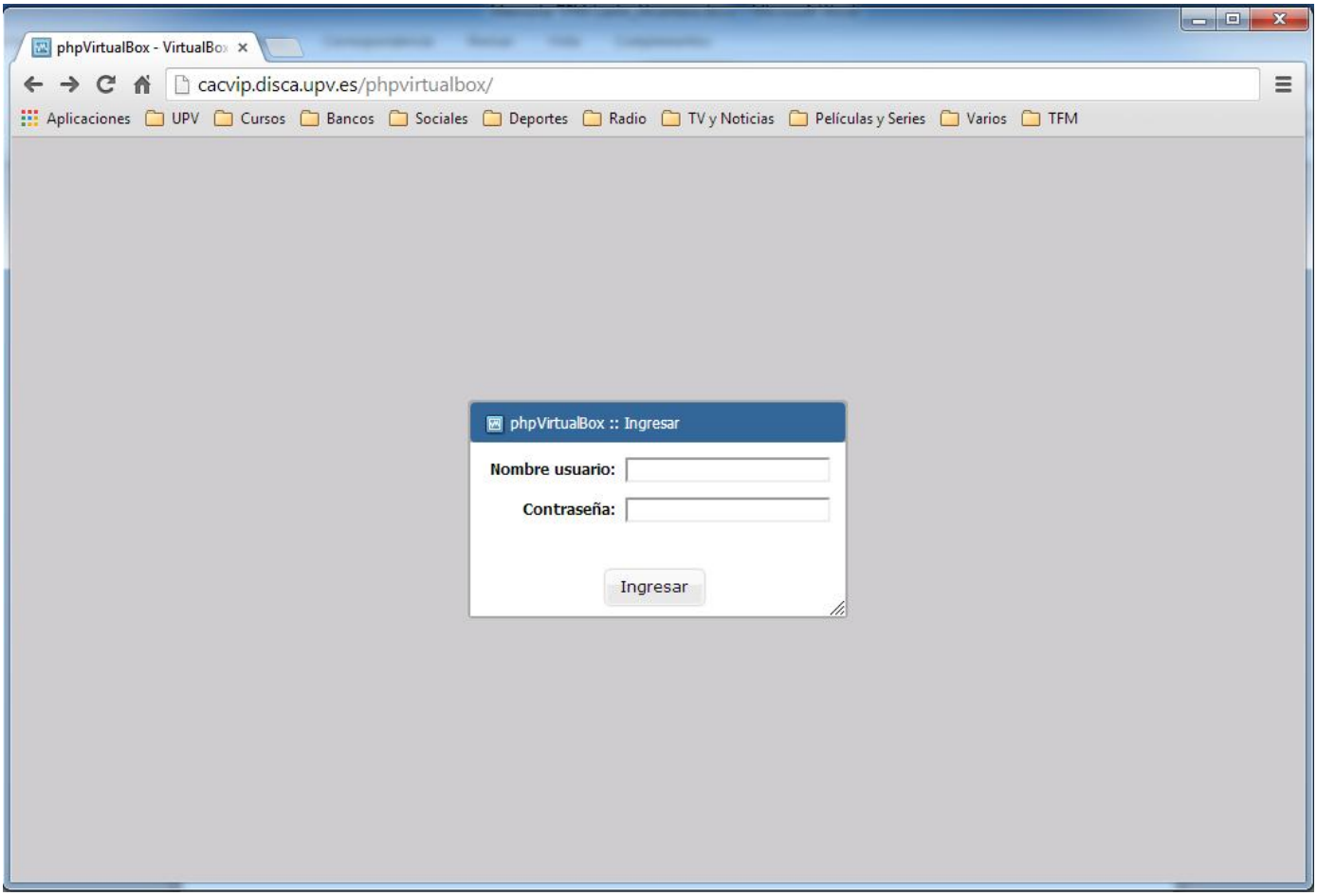
A continuación describimos los pasos o etapas que hemos seguido para probar y verificar el funcionamiento del sistema de máquinas virtuales:

1. **Acceso al servicio de máquinas virtuales del Servidor Master.**

Para realizar el acceso al servicio debemos introducir la siguiente URL en un navegador web: http://cac6.disca.upv.es/phpvirtualbox. Se nos abrirá la interfaz web de la herramienta PhpVirtualBox y nos pedirá un usuario y contraseña.

Por ejemplo: usuario = admin. Contraseña = admin. Este usuario es administrador por defecto de la interfaz web.

La siguiente figura muestra el aspecto de la pantalla de inicio.



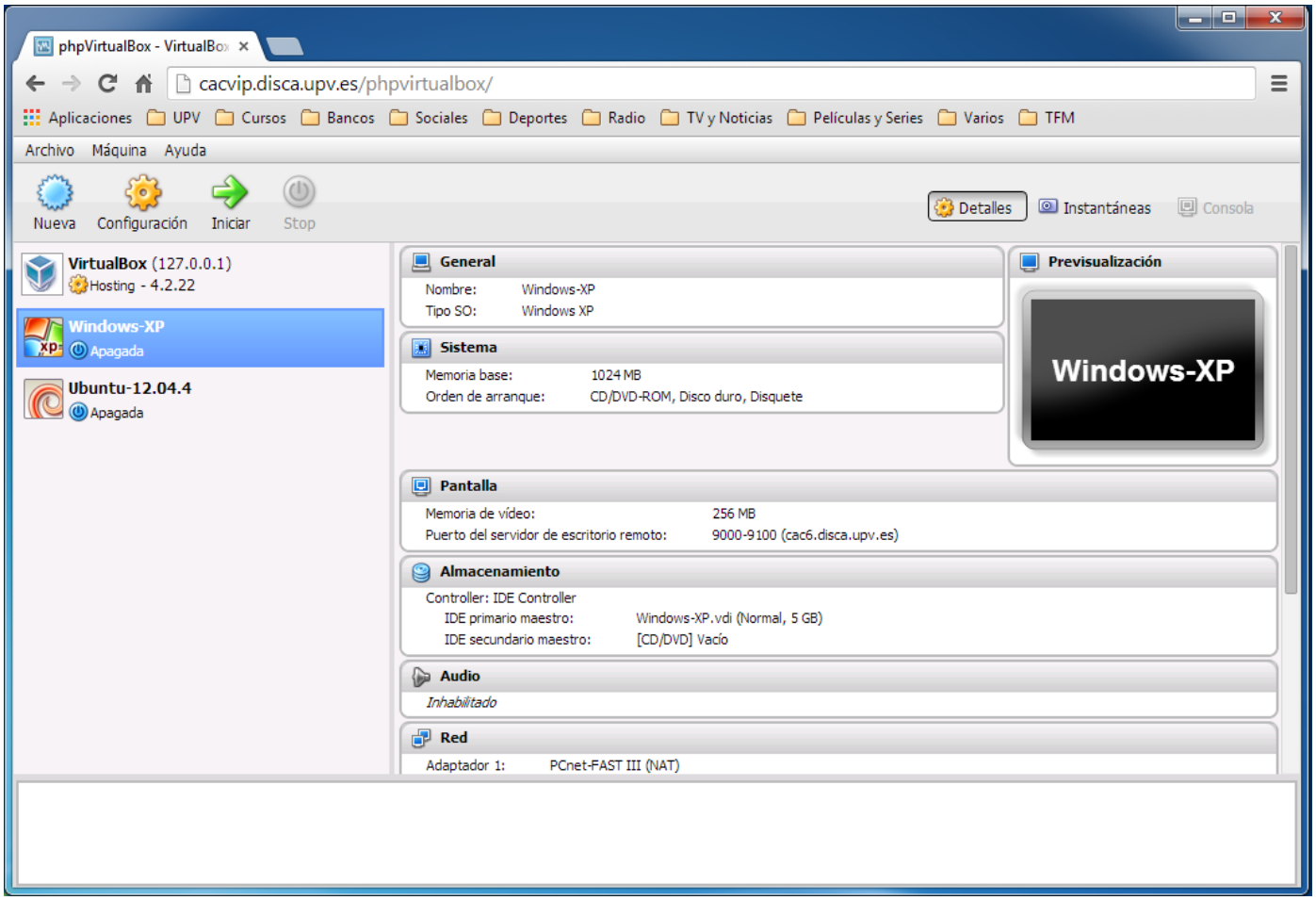
Pantalla de inicio PhpVirtualBox

2. Entorno de PhpVirtualBox.

El entorno web de PhpVirtualBox es idéntico al de Oracle VM Virtual Box. Desde esta interfaz web podemos realizar las siguientes tareas:

- Crear una nueva máquina virtual.
- Configurar una máquina virtual existente.
- Eliminar una máquina virtual.
- Arrancar máquinas virtuales.
- Crear y administrar usuarios de la interfaz web.

La siguiente figura muestra la interfaz web de PhpVirtualBox de la URL de nuestro servidor Master. A través de esta podemos administrar dos sistemas operativos virtualizados que se encuentran alojados en el Master, los cuales han sido creados desde la interfaz web de PhpVirtualBox. Actualmente el servidor ofrece el Sistema Operativo Windows XP y el Sistema Operativo Ubuntu 12.04.4.



Interfaz web de PhpVirtualBox

3. Inicio de las máquinas virtuales del servidor Master.

El Master tiene alojados dos sistemas operativos virtualizados mediante la herramienta Virtual Box. En concreto, los sistemas operativos son: Windows XP y Ubuntu 12.04.4. Desde la interfaz web PhpVirtualBox podemos ejecutarlas y acceder a dichos sistemas desde un ordenador remoto. A continuación mostramos ambos sistemas operativos ejecutándose mediante el protocolo RDP (Remote Desktop Protocol).

En la figura (a) podemos observar el Sistema Operativo Windows XP ejecutándose desde un ordenador remoto.

Figura (a) Sistema Operativo Windows XP

En la figura (b) apreciamos el Sistema Operativo Ubuntu 12.04.4 arrancado efectivamente desde un ordenador remoto al Cluster.

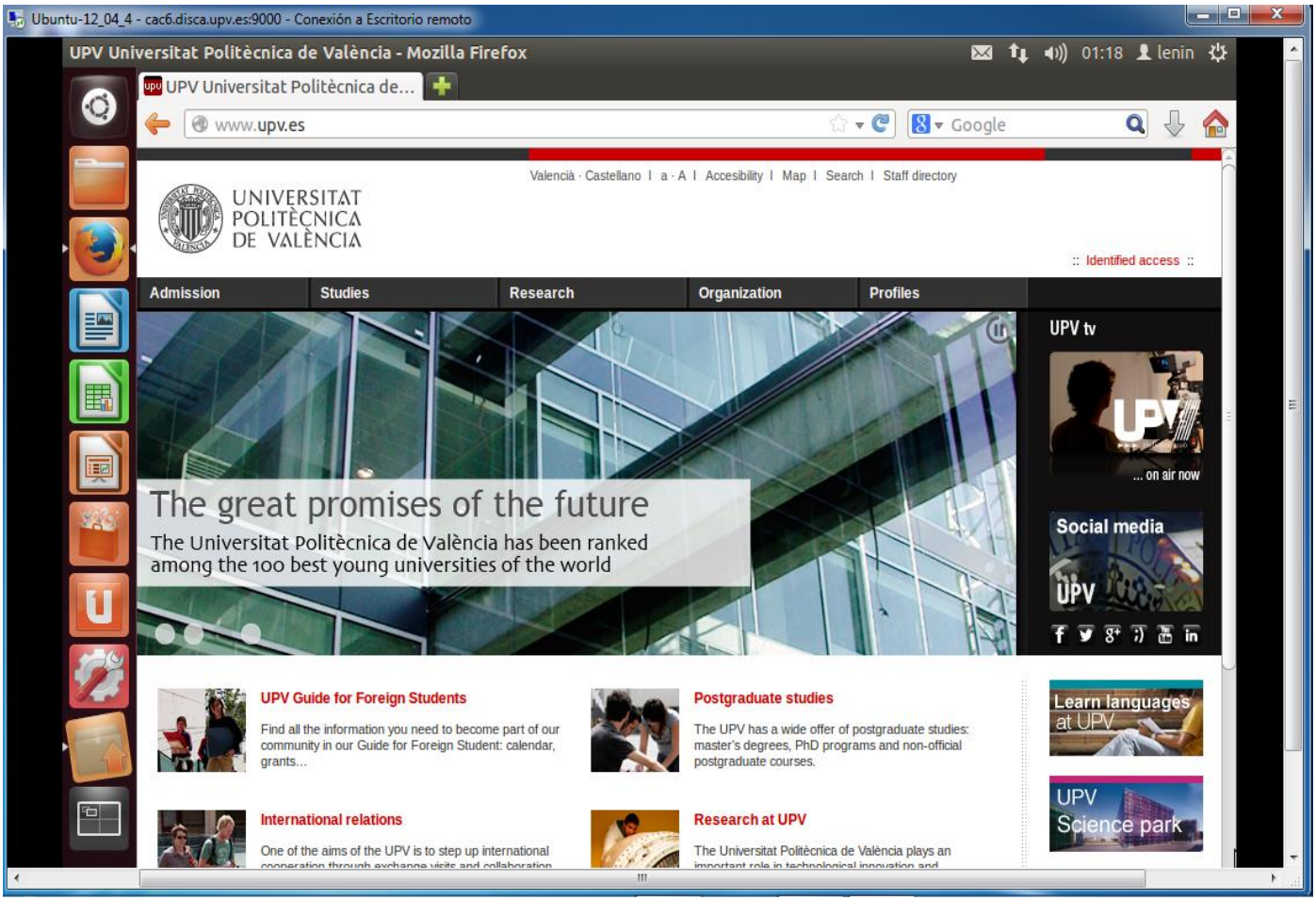


Figura (b) Sistema Operativo Ubuntu 12.04.4

4. Inicio de las máquinas virtuales en los nodos Esclavos.

Lo primero que realizamos fue la instalación del paquete Virtual Box en todos los nodos esclavos del Cluster desde el servidor Master empleando el script de utilidad *psh*, mediante la siguiente orden:

```
psh apt-get install linux-headers-$(uname -r) build-essential
virtualbox-4.2 dkms
```

De igual manera, instalamos el paquete Oracle VM Virtual Box Extension Pack mediante la orden siguiente:

```
psh VBoxManage extpack install
Oracle_VM_VirtualBox_Extension_Pack-4.1.18-78361.vbox-extpack
```

Creamos el usuario vbox, el cual será el encargado de ejecutar las máquinas virtuales en cada servidor y lo agregamos al grupo vboxusers. También creamos la contraseña de dicho usuario. Cabe destacar que este usuario es diferente al usuario de la interfaz.

```
useradd -m vbox -G vboxusers
```

```
passwd vbox
```

Copiamos el fichero */etc/default/virtualbox* desde nuestro servidor Master empleando la utilidad pscp.

```
pscp /etc/default/virtualbox
```

Una vez realizado esto, arrancamos el servicio web de Virtual Box mediante la siguiente orden:

```
psh update-rc.d vboxweb-service defaults
```

```
psh /etc/init.d/vboxweb-service start
```

Luego, desde el servidor Master debemos descargar la herramienta phpvirtualbox en el directorio compartido mediante nfs */websrv/www*. De esta manera los servidores esclavos pueden acceder al servicio.

```
cd /websrv/www
```

```
wget http://phpvirtualbox.googlecode.com/files/phpvirtualbox-
4.1-7.zip
```

Lo descomprimimos y configuramos tal y como hemos descrito en el apartado 4.7 de este escrito.

```
unzip phpvirtualbox-4.1-7.zip
```

Limitación:

Para acceder al servicio de máquinas virtuales en nuestros servidores esclavos tenemos una limitación. Esta consiste en que debemos acceder a nuestro nodo Master y lanzar desde esta un navegador web. Luego de esto ya podemos acceder a nuestras máquinas virtuales colocando en la barra de búsqueda del navegador lo siguiente:

```
192.168.1.210/phpvirtualbox
```

De esta manera podemos acceder a nuestro “Array” de servidores.

6. CONCLUSIONES

En los últimos años el Clustering se ha convertido en una pieza angular en el entorno tecnológico de organizaciones de diversas índoles. Implementar un Cluster es una técnica, que más allá de los beneficios que aporta, se ha convertido en algo necesario.

Empresas privadas, organizaciones públicas, universidades, y centros de investigación desde hace ya varias décadas se ha abocado a emplear Clusters para incrementar las prestaciones en los servicios que ofrecen, tanto de forma interna como externa.

Al finalizar esta implementación de un Cluster de Alta Disponibilidad con Equilibrado de Carga, hemos dado por sentada la frase que reza: “Mientras más, mejor”. Es decir, al tener a nuestra disposición una mayor cantidad de servidores configurados de forma homogénea y ofreciendo los mismos servicios, garantizamos mayores prestaciones y efectividad al momento de ofrecer nuestros servicios. De estas características se pueden beneficiar, tanto usuarios internos como usuarios externos.

Un Cluster con equilibrado de carga es bastante provechoso debido a que este tipo de Clusters toma la información de un servidor centralizado y la reparte por múltiples servidores. Estos servidores con carga equilibrada también se benefician del modelo HA (High Availability, Alta Disponibilidad). Este modelo introduce redundancia en todos los niveles. Un Cluster HA se beneficia en gran medida de tener una réplica de su Nodo Master. Es poco probable que todos los dispositivos duplicados fallen a la vez, exceptuando alguna catástrofe. Con la adición de un servidor extra este ayuda bastante en caso de fallo. A esta característica se conoce como redundancia **N+1**.

En este proyecto se ha instalado y configurado satisfactoriamente un cluster de alta disponibilidad con equilibrado (reparto) de la carga. Para hacer las tareas de instalación y mantenimiento se han personalizado un conjunto de herramientas que nos ayudan eficientemente a ejecutar estas operaciones sin necesidad de atender por completo todo el proceso.

Las pruebas realizadas en nuestro cluster han demostrado efectivamente que los objetivos planteados al inicio del proyecto se han llevado a cabo. Esto es, los resultados que han arrojado las pruebas y experimentos realizados sobre nuestro Cluster han servido como referencia para manifestar que el propósito definido ha sido alcanzado.

6.1. Trabajo Futuro.

Como trabajo futuro nos hemos planteado continuar provisionando el Cluster implementado con novedosas y ventajosas tecnologías que actualmente son de gran uso. Como un paso hacia el futuro es conveniente proveer al Cluster con el software MySQL Cluster para que este pueda tener incorporado un gestor de bases de datos.

Otro de los puntos de gran interés y que en futuras ampliaciones de este proyecto abordaremos, es el tema del control de energía del Cluster. Ya mucho se hablado del tema, he inclusive existen trabajos sobre este campo. Pero es de gran provecho implementar sobre el Cluster en cuestión un sistema que nos permita monitorizar y administrar el rendimiento energético del mismo.

7. BIBLIOGRAFÍA

1. **Clustering con Linux: Construcción y Mantenimiento de Clusters con Linux,** Charles Bookman. ISBN: 84-205-3771-3.

2. **Introducción a Condor,** Adrián Santos Marrero. http://www.iac.es/sieinvens/SINFIN/Condor/iac_manual/manual.pdf

3. **Bull Support On Line,** http://support.bull.com/

4. **NASDeluxe,** http://www.nasdeluxe.com/

5. **Top 500,** http://www.top500.org/statistics/list/

6. **Wikipedia,** http://www.wikipedia.org/

7. **Ubuntu,** https://help.ubuntu.com/community/

8. **Linux Virtual Server,** http://www.linuxvirtualserver.org/Documents.html#manuals

9. **Linux Virtual Server,** http://www.linuxvirtualserver.org/docs/scheduling.html

10. **Ldirectord,** http://www.noblenet.org/evergreenwiki/index.php/Ldirectord_setup

11. **Apache,** http://httpd.apache.org/

12. **Austin Tek,** http://www.austintek.com/LVS/LVS-HOWTO/

13. **Ubuntu Guía,** http://www.ubuntu-guia.com/

14. **Todo Programas,** http://www.todoprogramas.com/trucos/linux/

15. **How to Forge,** http://www.howtoforge.com/

16. **Virtual Box Images,** http://virtualboximages.com/

17. **Google Code,** https://code.google.com/p/phpvirtualbox/downloads/

18. **Virtual Box,** https://www.virtualbox.org/wiki/Downloads

19. **ArchLinux,** https://wiki.archlinux.org/index.php/PhpVirtualBox

20. **How Open Source,** http://www.howopensource.com/