

TRABAJO DE FINAL DE MÁSTER

INFRAESTRUCTURAS COMO SERVICIO EN ARQUITECTURAS DE NUBES PRIVADAS

Autor: David González Aragón

Tutor: Miquel Martín Mateo

Director: Toni Oller Arcas

Fecha: 11 de Junio de 2013

Índice

1.Introducción.....	3
2.Objetivos.....	4
3.Planificación del proyecto.....	6
Diagrama de Gantt.....	7
Diagrama de recursos del proyecto.....	7
4.Sistemas virtualizados y computación en la nube.....	8
Virtualización.....	8
Computación en la nube	9
5.Análisis de requerimientos y viabilidad.....	11
Escenario actual y problemática.....	11
Requisitos del sistema.....	11
6.Diseño de la arquitectura y del sistema.....	13
Comparativa entre plataformas de virtualización.....	13
VMware vSphere	13
Xen.....	13
KVM.....	15
OpenNebula.....	16
OpenStack.....	16
VirtManager.....	17
Elección del sistema de gestión de virtualización.....	19
7.Implementación y configuración del software de virtualización	20
Instalación del sistema de virtualización con Xen 4 en Debian Squeeze.....	20
Configurar múltiple interfaces de red.....	22
8.Implementación y configuración del sistema de gestión de MVs.....	23
Instalación del entorno web de VirtManager.....	23
Instalación del agente de gestión en las máquinas anfitrión.....	25
9.Resultados.....	26
VirtManager: Interfaz de usuario.....	26
10.Valoración económica	30
Planificación de recursos de personal.....	30
Impacto medioambiental y económico	31
11.Conclusiones.....	32
Trabajos futuros.....	32

1. Introducción

La Fundación i2CAT es un centro de investigación e innovación, que centra sus actividades en el desarrollo de la Internet del futuro. Sus principales objetivos son:

- Promover y fomentar el desarrollo tecnológico .
- Promover la investigación avanzada en el ámbito de las redes y las aplicaciones de banda ancha y potenciar su uso .
- Desarrollar tecnologías de Internet en el nivel de las aplicaciones, de los servicios y de la investigación de tecnología de red avanzada.
- Generar plataformas de colaboración entre el sector privado y el de investigación universitaria.
- Impulsar la colaboración con entidades e instituciones con objetivos similares.
- Colaborar y participar en proyectos de investigación nacionales e internacionales.

Dentro de la Fundación i2CAT existe el departamento de Infraestructuras y Servicio, donde se realiza este trabajo, el cual ofrece una base de infraestructura, servicios, software y know-how que permite a los diferentes departamentos disponer de un marco tecnológico y de conocimiento sólido donde poder realizar las tareas de investigación y desarrollo en el mejor contexto posible.

Dentro de este contexto, la gestión de los recursos de la infraestructura ha supuesto un problema cada vez más importante dentro de i2CAT, debido a que repercute en el coste del servicio por dos razones principales; el elevado consumo energético y el coste monetario de los servidores y los equipos necesarios para el correcto funcionamiento de los mismo.

Por este motivo la virtualización ha surgido como la solución para la consolidación de servidores, cuyo objetivo es conseguir un uso eficiente de los recursos para reducir el número total de servidores físicos y como consecuencia reducir los costes del consumo energético que presentaban los servidores.

Anteriormente estos servicios estaban basados en modelos de un servicio por equipo. Este esquema de un servicio por equipo dificulta la gestión de los equipos debido a la cantidad de sistemas operativos diferentes y el software instalado en ellos, pero la virtualización ha ofrecido una mejor utilización de los recursos de una máquina creando una capa de abstracción entre el hardware de la máquina física y el sistema operativo de la máquina virtual, el cual es totalmente independiente lo que permite que los posibles errores de funcionamiento de la máquina virtual no provoquen efectos colaterales a las demás.

En base a estos principios, el objetivo de este trabajo es proporcionar una arquitectura de nube privada basada en sistemas virtualizados, desarrollando el concepto de IaaS¹, donde la arquitectura propuesta ofrece un nuevo modelo de servicio orientado especialmente a infraestructura, que a su vez permita ofrecer el servicio de escritorios y servidores virtualizados al personal de la Fundación i2CAT.

1 [Infrastructure as a Service](#)

2. Objetivos

Los objetivos que se han propuesto inicialmente dentro del trabajo son los siguientes:

1. Estudio de las condiciones iniciales y estado actual del sistema:

Antes de emprender el trabajo es necesario establecer los conceptos básicos en el cual se fundamenta la arquitectura del sistema y las condiciones iniciales para poder implementar el sistema gestor de recursos o IA.

2. Transparencia y facilidad de uso:

Los cambios de configuración han de ser transparentes al usuario final. Para ello es importante implementar una interfaz en la que permita al usuario final gestionar los recursos iniciales y características de sus MVs.

3. Consolidación de servidores:

La consolidación de servidores trata de definir un conjunto de estrategias para reducir costes asociados con el funcionamiento del conjunto de servidores. Tiene como objetivo simplificar la gestión de recursos, reducir la complejidad y la carga de trabajo de los servidores.

4. Aprovechamiento óptimo de los recursos y ahorro de energía:

El consumo energético de los centros de datos esta aumentando a medida que aumentan los servicios de computación en la nube. Esto supone un importante derroche de energía y costes en proporción a las dimensiones que presente el centro de datos. Debido a los costes que supone, el ahorro de energía es una de las prioridades en el diseño de arquitecturas virtualizadas.

Un de los principales motivos de este exceso de consumo energético en los centros de datos se deben principalmente a la infrautilización del hardware disponible.

Por este motivo, el impacto que supone la virtualización sobre el consumo energético es muy significativo, además al necesitar menos hardware también se ahorra en la compra de servidores y los equipos necesarios para el correcto funcionamiento de los mismos.

5. Migración de máquinas virtuales:

Una de las características que ofrece la arquitectura es la habilidad de mover máquinas virtuales (MVs) de una máquina física a otra de manera transparente.

6. Escalabilidad:

El número N de servidores no ha de tener cota, viene definido únicamente por las necesidades totales.

7. Monitorización:

Para poder gestionar los recursos del sistema se requiere implementar la capacidad de monitorizar los recursos tanto de los nodos secundarios en tiempo real y de forma centralizada.

8. Por último el principal objetivo de este trabajo es implementar la plataforma en el entorno de trabajo real de la Fundación i2CAT.

3. Planificación del proyecto

El proyecto se ha planificado en las siguiente fases y fechas:

- **Análisis de requerimientos y viabilidad del proyecto, del 14 al 20 de Marzo:**
En esta fase se recogerán los requisitos del proyecto tanto técnicos como a nivel de usuario y se analizará la viabilidad del proyecto, analizando si existen las tecnologías actuales que puedan cumplir con los requisitos propuestos.
- **Diseño de la arquitectura y del sistema, de 21 al 28 de Marzo:**
En esta fase se diseñará la arquitectura y el sistema para que cumpla con los requisitos recogidos en la fase anterior y con los objetivos del proyecto. Se buscarán las tecnologías que cumplan con los objetivos y se analizarán las diferentes soluciones.
- **Instalación y configuración del sistema, del 29 de Marzo al 10 de Abril:**
En esta fase se instalará el software elegido en el diseño y se configurará tanto la arquitectura del sistema como las diferentes herramientas, en las cuales se incluyen el sistema operativo, software de virtualización, etc.
- **Instalación y configuración del sistema de gestión de MVs, del 11 al 23 Abril:**
En esta fase se instalará y configurará una herramienta que permita crear las maquinas virtuales a los usuarios de forma rápida y sencilla.
- **Implementación de sistema de escritorio remoto en microordenadores, del 24 de Abril al 1 de Mayo:**
En esta fase se instalará y pondrá a prueba el funcionamiento de escritorios virtuales desde microordenadores.
- **Pruebas de funcionamiento, rendimiento y resultados, del 2 al 17 de Mayo:**
En esta fase se harán diferentes pruebas de funcionamiento para poder analizar el resultado del proyecto, su funcionamiento y el rendimiento del mismo.
- **Redacción memoria de proyecto, del 1 de Mayo al 11 de Junio:**
En esta fase se redactará la memoria final de los pasos realizados en el proyecto.
- **Preparación de la presentación del proyecto, del 2 al 17 de Mayo:**
Durante esta fase se creara una presentación del proyecto para defenderlo ante el tribunal.

Diagrama de Gantt

En el siguiente diagrama de Gantt se puede ver las tareas y la planificación del proyecto.

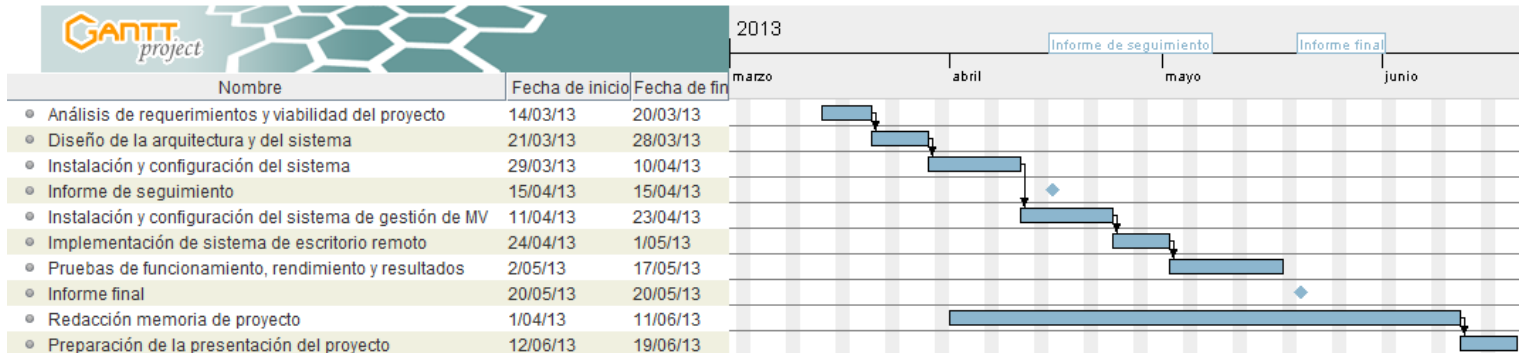


Imagen 1: Diagrama de Gantt

Diagrama de recursos del proyecto

En el diagrama de recursos podemos distinguir los recursos de personal asignados:

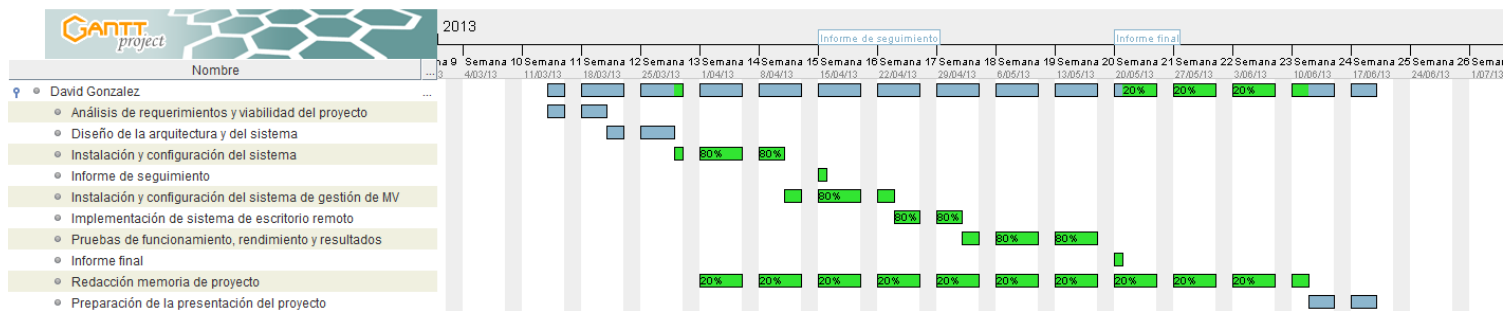


Imagen 2: Diagrama de recursos

4. Sistemas virtualizados y computación en la nube

Virtualización

Virtualización es un término que se refiere a la abstracción de los recursos de una máquina. Esta definición se aplica a todo tipo de equipos e infraestructura informática, como pueden ser recursos de red, plataformas hardware, sistemas operativos o dispositivos de almacenamiento.

En el caso de virtualización de servidores esta capa de abstracción, más conocida como Hypervisor, se encarga de manejar y gestionar los recursos de hardware principales, CPU, disco, memoria RAM, etc. Por tanto, Hypervisor es el encargado de repartir estos recursos entre las diferentes máquinas virtuales que se estén ejecutando sobre la misma máquina física.

La virtualización ofrecen grandes mejoras y ventajas respecto a un sistema de hardware nativo:

- **Ahorro:** Mejora substancialmente el aprovechamiento de los recursos de hardware, lo que se traduce en un gran ahorro en cuanto a coste del hardware y también un ahorro en cuanto a energía consumida y espacio.
- **GreenIT²:** Al reducir el número de máquinas, reducimos la cantidad de energía utilizada, no sólo en alimentar el servidor sino que también se reduce el consumo de refrigeración tanto eléctrico como las emisiones de aire caliente.
- **Flexibilidad:** En una misma máquina podemos tener funcionando juntos un sistema operativo Windows y un sistema basado en GNU/Linux .
- **Escalabilidad y tiempo de funcionamiento:** La virtualización ofrece un sistema totalmente homogéneo a las máquinas virtuales.

Dependiendo del hardware y del sistema operativo podemos distinguir hasta cuatro tipos de virtualización diferentes:

- **Emulación o simulación:** el sistema operativo padre simula un hardware completo, esta simulación hace que el rendimiento de la máquina virtual sea bastante menor.
- **Virtualización a nivel de Sistema Operativo:** Algunas comunidades no lo incluyen dentro de la definición de virtualización como tal, porque se basa en crear celdas de usuarios independientes, sin acceso entre ellas pero que comparten el mismo sistema operativo y aplicaciones.
- **Paravirtualización:** la máquina virtual no simula un hardware, sino que ofrece una interfaz al sistema que sólo puede usarse mediante la modificación del kernel del sistema operativo de la máquina virtual. . La paravirtualización se utiliza únicamente con sistemas operativos basados en GNU/Linux y BSD, ya que estos permiten la modificación del kernel, en el caso de Windows y MAC OSX no es posible ya que al ser propietario no es posible estas modificaciones.

2 http://es.wikipedia.org/wiki/Green_computing

- **Virtualización nativa o completa:** la máquina virtual simula un hardware subyacente completo para permitir instalar un sistema operativo sin modificar en la máquina virtual. La Máquina Virtual desconoce la existencia del entorno virtualizado y como no requiere ser modificado es posible instalar cualquier sistema operativo compatible con el hardware.

Los elementos principales de un sistema virtualizado son:

1. **Hypervisor:** El Hypervisor actúa como una capa que gestiona el hardware, virtualizándolo y haciendo que estos recursos físicos sean compartidos y usados de forma transparente por múltiples sistemas operativos. Las instancias de estos sistemas operativos son las denominadas máquinas virtuales que se describen a continuación.
2. **Anfitrión:** Es el sistema operativo principal a partir del cual se implementan las diferentes particiones o MVs mediante el Hypervisor y la asignación de recursos.
3. **Máquina Virtual (MV):** Es la máquina cliente o invitada que funciona sobre el sistema operativo Anfitrión. Proporciona aislamiento total del resto de MVs que corren sobre Anfitrión y es transparente al usuario final.

Computación en la nube

La computación en la nube es un compendio de servicios brindados a través de la red, los cuales pueden dar acceso tanto a hardware y recursos como aplicaciones, almacenamiento, ancho de banda, todos ellos integrados en una plataforma que es transparente al usuario final. La computación en la nube utiliza la virtualización como clave para su desarrollo.

Las ventajas que aporta estas arquitecturas son diversas, pero las que hacen que sean un factor clave en el mundo IT actual son su elasticidad y escalabilidad, que permite variar la capacidad de los recursos brindados a los usuarios en función de la demanda.

Por otra parte la computación en la nube brinda una ventaja de auto aprovisionamiento, permitiendo a los usuarios obtener los recursos o servicios automáticamente evitando los retrasos que supone el modelo tradicional de centro de datos. También podemos resaltar la accesibilidad, pues permite el uso de los recursos independientemente del dispositivo que posee el usuario final.

La computación en la nube se puede diferenciar en varias capas o niveles, en el nivel inferior tenemos los recursos físicos, que es básicamente el hardware que utiliza, encima del cual se instala la virtualización para poder repartir estos recursos físicos a petición del software que lo gestiona.

Encima de estas capas podemos diferenciar los diferentes modelos de computación y servicio en la nube basados en los servicios que soporta:

1. Infraestructura como servicio (IaaS³):

Los recursos físicos, tales como los servidores, los dispositivos de red o almacenamiento se ofrecen como servicio. También incluye el sistema operativo y la tecnología de virtualización para gestionar los recursos. Un ejemplo de este modelo es Amazon EC2.

3 Del inglés Infrastructure as a Service

2. Plataforma como servicio (PaaS⁴):

Con este modelo se provee el conjunto de software necesario para que el usuario desarrolle su propia solución. Un ejemplo de PaaS es el hosting web.

3. Software como servicio (SaaS⁵):

En este modelo el software final y los datos que maneja los aloja un tercero, el cual provee el acceso a este software mediante un navegador web, un cliente de escritorio o móvil.

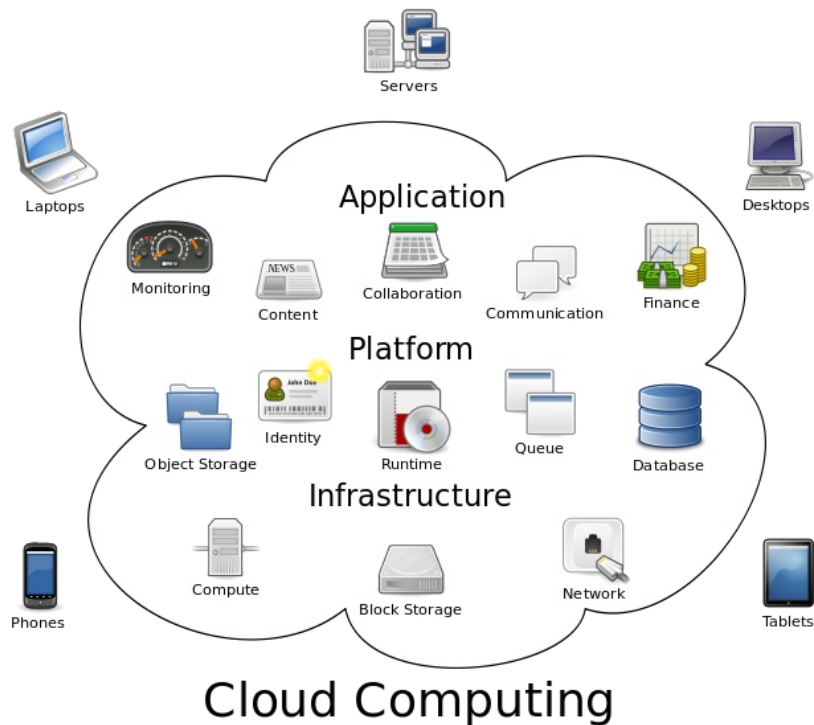


Imagen 3: Computación en la nube basada en servicio

4 Del inglés Platform as a Service
 5 Del inglés Software as a Service

5. Análisis de requerimientos y viabilidad

Escenario actual y problemática

Actualmente la Fundación i2CAT dispone de unos 30 servidores físicos virtualizados que alojan un número aproximado de 180 Máquinas Virtuales.

Estos servidores tienen instalado el sistema operativo Debian Squeeze y utilizan la plataforma de virtualización de Xen Source en su versión 4 para crear, configurar y gestionar las Máquinas Virtuales.

Cada uno de estos servidores se gestiona de forma individual utilizando las herramientas de Xen instaladas en cada uno de los servidores y no existe ninguna herramienta en funcionamiento que los gestione de forma centralizada y global. A pesar de esto existen protocolos y configuraciones genéricas para todos los servidores de forma que su instalación y configuración esta estandarizada, de esta forma todos los sistemas operativos anfitriones comparten características similares de forma que todas las máquinas virtuales son compatibles entre los diferentes anfitriones.

Esta falta de control central genera problemas en la administración y gestión de los servidores:

- Hay poco control sobre las Máquinas Virtuales existentes, cuales están en funcionamiento y cuales no.
- La creación de una nueva Máquina Virtual se ha de hacer de forma manual accediendo al servidor.
- El personal de la Fundación no tiene ninguna opción para crear máquinas virtuales bajo demanda sin tener que pasar a través del departamento de Infraestructuras y Servicios.
- Se tiene poco conocimiento de la carga de trabajo de cada una de las máquinas virtuales y los anfitriones, siendo necesario acceder a la máquina para comprobar el estado.

El objetivo de este trabajo es implementar una plataforma que solucione los problemas mencionados y que cumpla con los siguientes requisitos:

Requisitos del sistema

La plataforma de virtualización ha de cumplir diferentes requisitos:

1. Facilidad de uso por parte del personal: El acceso debe ser mediante un navegador, con formularios intuitivos y sin complicaciones a la hora de crear o gestionar sus máquinas.
2. Integración con el sistema de identificación única: La Fundación i2CAT dispone de un sistema de identificación única y gestión de roles basado en un directorio LDAP central donde se almacenan y gestionan todos los perfiles. La plataforma debe permitir su integración con este directorio.

3. Usuarios, grupos y roles: La plataforma debe permitir gestionar los diferentes usuarios, roles y departamentos a los que pertenece permitiendo una gestión eficiente de las máquinas virtuales y que el personal solo tenga acceso a la gestión de sus propias máquinas virtuales
4. Gestión de sistema operativo anfitrión: la plataforma debe ser capaz de añadir nuevos sistemas operativos anfitriones, una vez añadidos debe ser posible gestionarlos y crear Máquinas Virtuales sobre ellos.
5. Imágenes de máquinas virtuales: La plataforma debe disponer de imágenes prediseñadas de los diferentes sistemas operativos invitados para la rápida instalación de los mismos. Además de un sistema para añadir estas imágenes a la plataforma. De esta forma el personal tendrá libre elección de sistema operativo.
6. Libre elección de los recursos del sistema: El personal debe poder disponer de la cantidad de memoria RAM, vCPUs, capacidad del disco duro y interfaces de red necesarias para el correcto funcionamiento de la Máquina Virtual. Por otra parte la plataforma dispondrá de un sistema para evitar que los usuarios puedan crear Máquinas Virtuales que superen el límite de recursos disponible.
7. Monitorización de recursos: La plataforma debe tener la capacidad de monitorizar el estado de los recursos, las máquinas virtuales y los sistemas operativos anfitrión de forma que se puedan mostrar y controlar estos recursos.
8. Resistencia a fallos de hardware: La plataforma debe ser resistente a fallos de hardware de forma que las máquinas virtuales se puedan migrar a otras de forma fácil y rápida.
9. Aprovechamiento óptimo de los recursos: La plataforma ha de ser capaz de exprimir los recursos de hardware disponibles para su optimización de uso.
10. Sin limitaciones de crecimiento de hardware y necesidades de los usuarios: La plataforma no puede estar limitada en crecimiento y debe disponer de la posibilidad de añadir más servidores de forma fácil y rápida.
11. Gestión de redes virtuales: La plataforma ha de ser capaz de gestionar las diferentes redes que dispone la Fundación y ofrecer a cada máquina virtual
12. Fácil de mantener, administrar y que suponga el coste menor posible para la Fundación i2CAT: En la medida de la posible se deben utilizar tecnologías basadas en software libre que nos permitan modificar las y que no supongan una inversión mayor en la plataforma.

6. Diseño de la arquitectura y del sistema

Comparativa entre plataformas de virtualización

VMware vSphere

Este sistema es un conjunto de varias herramientas que cubren todas las necesidades de una arquitectura de nube privada, entre los cuales diferenciamos: Por una parte vCenter la cual es la aplicación que de forma centralizada ofrece el control de las máquinas virtuales a los administradores del sistema.

Por otra parte VMware ESX Server es un sistema complejo de virtualización, el se ejecuta como sistema operativo propio (basado en Red Hat Enterprise Linux) dedicado al manejo y administración de máquinas virtuales, lo que implica que no necesita un sistema operativo externo sobre el cual instalarlo.

Para su administración, hay que instalar un software en una máquina remota, conectándose mediante una interfaz web.

Soporta virtualización nativa y paravirtualización mediante un hypervisor propio de VMware, incluye soporte para SMP avanzado, rendimiento muy aproximado al nativo, documentación abundante y un buen soporte técnico, pero de pago.

Dispone de una versión gratuita llamada ESXi la cual esta limitada a un máximo de 32Gb de memoria RAM por servidor.

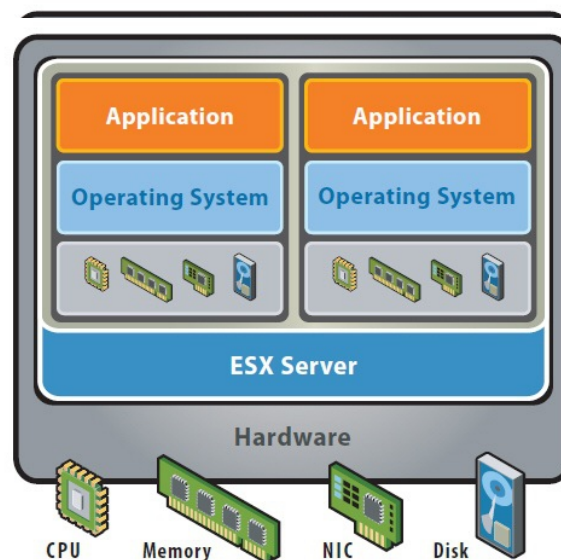


Imagen 4: Virtualización con VMWare

Xen

Xen es una plataforma de virtualización de código abierto desarrollada originalmente por Ian Pratt de la Universidad de Cambridge en 2003. La meta del diseño era poder

ejecutar instancias de sistemas operativos con todas sus características, de forma completamente funcional en un equipo sencillo.

Posteriormente Ian Pratt fundó la compañía XenSource, Inc. la cual fue comprada a finales de octubre del 2007 Citrix Systems, compañía especializada en software de infraestructuras de acceso.

Desde su compra Xen se ha dividido en dos versiones, una comercial y otra de software libre:

Citrix CloudPortal

Esta versión comercial esta compuesta por varias herramientas web, que permiten la administración de las máquinas virtuales de forma remota y centralizada, y XenServer, el cual permite la virtualización de servidores mediante el hypervisor de Xen.

Xen Source

Versión de software libre, con una gran comunidad de desarrolladores y usuarios.

Existen varios proyectos creados por la comunidad de XenSource entre ellos podemos distinguir XCP (Xen Cloud Platform) que permite la administración web centralizada de las máquinas de forma remota y Xen hypervisor, el cual se encuentra actualmente en la versión 4.2.1 y entre sus funcionalidades se encuentran:

- Soporte de paravirtualización.
- Soporte de virtualización nativa.
- Migración de maquinas virtuales en tiempo real.
- Rendimiento en paravirtualización y virtualización nativa cercano al de una máquina física.

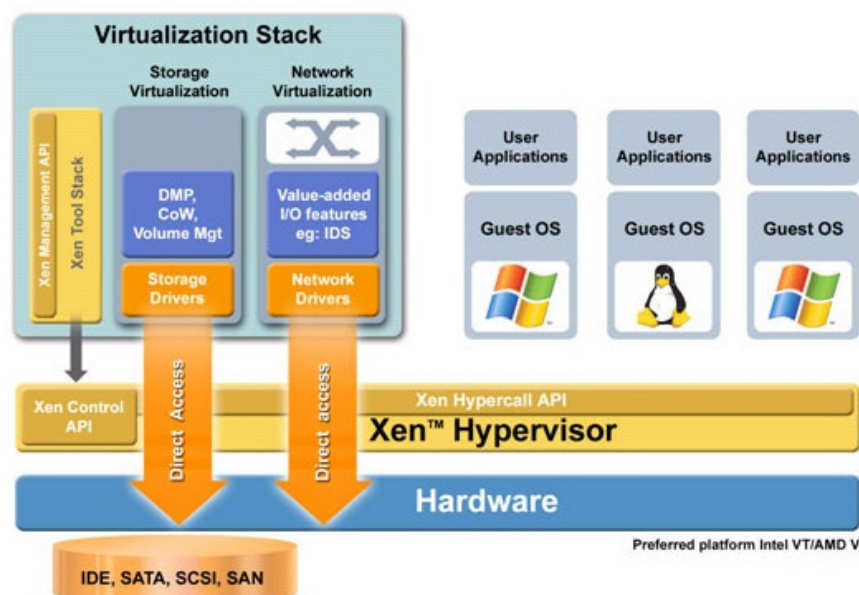


Imagen 5: Plataforma de virtualización Xen

KVM

Kernel Virtual Machine, creado por Qumranet, es una solución para implementar virtualización nativa con Linux, sobre hardware x86. Está formada por un módulo del núcleo y herramientas en el espacio de usuario, siendo en su totalidad software libre.

KVM utiliza una versión modificada de QEMU como interfaz de usuario y permite ejecutar máquinas virtuales utilizando imágenes de disco que contienen sistemas operativos sin modificar. Por contra, KVM sólo soporta virtualización nativa lo que limita su uso a un hardware específico (Intel VT o AMD-V).

KVM no dispone de un sistema propio de administración de las maquinas virtuales por lo que hay que instalar uno adicional.

Tabla comparativa entre las diferentes soluciones de virtualización:



Versión	5	4	4.1.2
Edición	Essentials	-	-
Empresa	Vmware	Red Hat	Xen.org
Licencia	Propietaria	Software libre	Software libre
Tipo de licencia	Por servidor	-	-
Mercado objetivo	Gran empresa	Personal PYME	Gran empresa
Especificaciones técnicas			
Hypervisor	BareMetal (Sistema operativo propio)	Anfitrión	Anfitrión
Tipos de virtualización soportados	Virtualización completa. Virtualización a nivel de Sistema Operativo. Paravirtualizacion.	Virtualización completa.	Virtualización completa. Paravirtualizacion.
Arquitecturas soportadas	x86 x64 DAS eSATA	x86 x64	x86 x64
Modos de acceso a disco	FC FCoE iSCSI NAS RDM SSD for Swap USB	DAS eSATA FC iSCSI NAS USB	DAS eSATA FC iSCSI NAS USB
Limitaciones			
CPUs por VM	32 vCPUs	-	-
RAM por VM	1,000 GB	-	-
Tamaño de discoVirtual	2000 GB	-	-
VMs en cada anfitrión	512 VMs	-	-

RAM en el anfitrión	32 GB	-	-
Compatibilidad y soporte			
Sistemas operativos anfitrión soportados	-	CentOS Fedora Red Hat Linux Ubuntu Debian	CentOS SUSE Linux Ubuntu Debian
Soporte	FAQ Foros	FAQ Foros Online Papers	FAQ Foros Papers

Entre las herramienta de gestión de nubes privadas consolidadas como software libre podemos destacar OpenNebula y OpenStack como dos de las proyectos de software libre más consolidados y usados :

OpenNebula

OpenNebula es una herramienta de software libre que permite construir cualquier tipo de cloud: privado, público e híbrido. Ha sido diseñado para ser integrado con cualquier tipo de red y almacenamiento, para así adaptarse a cualquier centro de datos existente proporcionando una plataforma elástica, que ofrece un rápido despliegue y escalabilidad de maquinas virtuales.

OpenNebula implementa gran cantidad de funcionalidades, entre las que diferenciamos:

- Gestión de usuarios y roles.
- Gestión de los nodos.
- Gestión de las imágenes de máquinas virtuales
- Gestión de clústers de maquinas físicas.
- Gestión de las redes virtuales.

OpenStack

OpenStack es un conjunto de herramientas de software libre que facilitan la implantación de una arquitectura de nube privada fácilmente escalable. Las principales características que ofrece OpenStack son:

- On-demand Self-service, los usuarios pueden acceder a los servidores y redes de forma autónoma.
- Resource Pooling, los usuarios pueden disponer de más recursos en función de sus necesidades y la demanda global.
- Elasticidad, facilidad de acceso bajo demanda a los recursos de forma rápida y escalable.
- Servicio a medida, optimización el uso de recursos y control en nivel de servicio de los recursos.

VirtManager

VirtManager es una herramienta web de gestión de servidores destinados a la virtualización. Entre sus funcionalidades se encuentran la creación de máquinas virtuales mediante paravirtualización o virtualización nativa, el mantenimiento de éstas o la asignación de responsables.

Esta desarrolla íntegramente por el personal de la Fundación i2CAT y deriva principalmente del trabajo y conocimiento realizado dentro ámbito del proyecto europeo Ofelia⁶.

La arquitectura general del VirtManager esta dividida en cinco bloques principales:

- Web/Application Apache Tomcat Server: es el servidor de aplicaciones web que despliega el servicio web.
- Virtualization Server (Agente): es el servidor o servidores anfitriones en el cual se crean las máquinas virtuales. En esta versión de VirtManager, solo se soporta Xen como software de virtualización.
- Database: es la base de datos del sistema, que en nuestro caso, ha sido implementada en MySQL.
- LDAP: Es el servidor que contiene el directorio de nombres y organizaciones de la Fundación i2CAT. Se utiliza para autenticar a los usuarios y obtener sus permisos de usuario, este servidor ya esta implementado y en funcionamiento.
- UserComputer: Es el usuario final, que desde su navegador web, es capaz de acceder al sistema de gestión de virtualización.

6 <http://www.fp7-ofelia.eu/>

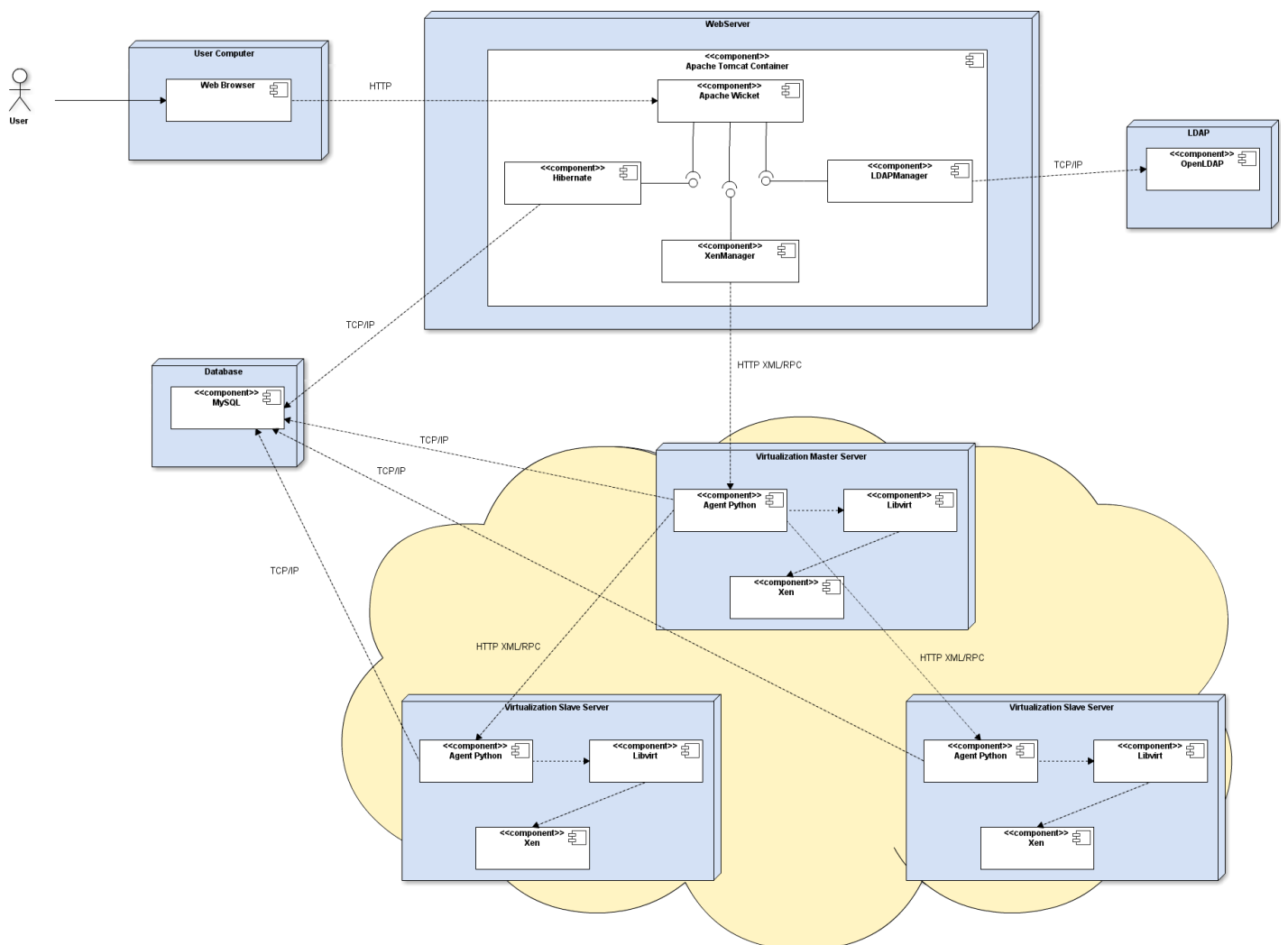


Imagen 6: Esquema general de VirtManager

Aplicación web

La aplicación web ofrece una serie de funcionalidades definidas previamente, como son el acceso de usuario o la creación de máquinas virtuales. Para ofrecer estas funcionalidades, VirtManager se compone de diferentes funciones que permitan interactuar con el usuario.

Home: Página principal de la aplicación desde donde se tiene acceso al resto de páginas del sitio web.

- CreateVM: Permite la creación de máquinas virtuales mediante un formulario.
- EditVM: Permite la edición de algunos parámetros de una máquina virtual previamente creada.
- ListMachines: Página dónde se muestran todas las máquinas virtuales previamente creadas, junto con información importante: nombre, tamaño de disco duro, tamaño de memoria, etc.
- AddHost: Permite añadir nuevos servidores anfitrión donde virtualizar nuevas MVs.
- AddOS: Permite añadir nuevos sistemas operativos al sistema para ser virtualizados.

- **MonitorResources:** Permite monitorizar el uso de CPU y RAM de los diferentes nodos del clúster.

Agente de VirtManager

El agente de VirtManager es una herramienta desarrollada en Python que permite controlar a los diferentes anfitriones que forman el sistema de virtualización. Se encarga de recibir las órdenes enviadas desde la web de VirtManager, procesarlas y ejecutarlas.

Además, recopila información de monitorización de cada uno de los nodos y las almacena en la base de datos central. De esta manera desde la web se podrá saber cuál es el estado de cada uno de ellos.

Todas las comunicaciones entre agentes se establecen vía HTTP XML/RPC. Mientras que las conexiones a la base de datos se establecen vía TCP/IP.

Base de datos

En la base de datos se almacenan toda la información referente al sistema de gestión de MVs, desde la información disponible en la aplicación web, la asignación y los datos de las diferentes MVs y servidores anfitriones. También se almacena en la base de datos la monitorización de las MVs y los servidores anfitriones que las alojan, esta información la envían directamente los agentes de VirtManager sin pasar por la aplicación web.

Servidor LDAP

El LDAP es el directorio de usuario y grupos de la Fundación i2CAT, el cual permite autenticar de forma centralizada a los usuarios y gestiona sus permisos, este servidor ya está implementado y en funcionamiento, para facilitar la gestión de las MVs es necesario la correcta integración con el directorio LDAP.

Elección del sistema de gestión de virtualización

Los motivos por los que desde la dirección de i2CAT se ha decidido implementar y utilizar VirtManager son:

- Cumple los requisitos de gestión de usuarios, principalmente la integración con el directorio LDAP y los grupos de usuarios.
- Es un software de gestión propio, adaptado a las necesidades de la fundación.
- Es un proyecto ya realizado, que funciona y está en producción.
- Los cambios que han de ser implementados son rápidos y sencillos.
- La integración en producción relativamente sencilla y no es necesario una re-instalación de los sistemas, solo instalar nuevos componentes.

7. Implementación y configuración del software de virtualización

Instalación del sistema de virtualización con Xen 4 en Debian Squeeze

Primero de todo configuramos el apt para que contenga todos los paquetes necesarios, editando el archivo `/etc/apt/sources.list` y modificando lo de forma acorde a:

```
deb http://ftp.es.debian.org/debian/ squeeze main contrib non-free
deb-src http://ftp.es.debian.org/debian/ squeeze main contrib non-free
deb http://security.debian.org/ squeeze/updates main
deb-src http://security.debian.org/ squeeze/updates main
#squeeze-updates, previously known as 'volatile'
deb http://ftp.es.debian.org/debian/ squeeze-updates main
deb-src http://ftp.es.debian.org/debian/ squeeze-updates main
```

A continuación actualizamos la base de datos del apt:

```
apt-get update
```

Ejecutamos el script `tuneAserver`, script propia de la Fundación i2CAT, el cual automatiza todo los cambios necesarios en el servidor.

```
wget http://calisto.i2cat.net/tuneAserver && chmod +x tuneAserver
./tuneAserver
```

Elegimos la opción a y vamos instalando todos los paquetes y modificaciones necesarias, una vez instalado todo podemos borrar el script.

Instalamos Xen mediante el apt instalando los paquetes necesarios:

```
apt-get install xen-hypervisor-4.0-amd64 xen-linux-system-2.6-xen-amd64 xen-
utils-4.0 xenstore-utils xenwatch xen-tools xen-qemu-dm-4.0
```

Editamos el archivo `/etc/xen/xend-config.sxp` y descomentamos la línea (`network-script network-bridge`) y (`vnc-listen '127.0.0.1'`) modificando el archivo para que quede:

```
(network-script network-bridge)
[...]
(dom0-min-mem 512)
[...]
(vnc-listen '0.0.0.0')
```

Cambiamos el orden de arranque del grub2 y lo actualizamos.

```
mv /etc/grub.d/10_linux /etc/grub.d/50_linux
update-grub2
```

En caso de que la virtualización necesite usar más de una interfaz configuraremos y usaremos el script de `multibrige`, el cual se explica en el punto siguiente.

A continuación reiniciamos la maquina.

Una vez arranque de nuevo, comprobamos que el kernel sea el correcto:

```
uname -a
```

```
Linux saturno 2.6.32-5-xen-amd64 #1 SMP Wed Jan 12 05:46:49 UTC 2011 x86_64  
GNU/Linux
```

Montamos la carpeta compartida por red en el directorio /images añadiéndola primero al fstab con el comando:

```
mkdir /images && echo "nfs.i2cat.net:/images /images nfs ro,sync,hard,intr 0  
0" >> /etc/fstab
```

A continuación montamos la carpeta compartida por red:

```
mount /images
```

Creamos el volume group vservers de LVM⁷, el device puede cambiar y es necesario usar una o varias particiones libres y sin formatear, en este caso usamos un device de RAID 1 llamado md2.

```
pvcreate /dev/md2
```

```
vgcreate vservers /dev/md2
```

Por último editamos el archivo /etc/xen-tools/xen-tools.conf

```
lvm = vservers
```

```
[...]
```

```
install-method = tar
```

```
[...]
```

```
install-source = /images/debian-squeeze-xen4.tar
```

```
[...]
```

```
dhcp = 1
```

```
[...]
```

```
tar-cmd = /bin/tar --numeric-owner -xvf $src
```

```
[...]
```

```
genpass = 0
```

```
[...]
```

```
passwd = 1 #Para que te pregunte el password.
```

```
[...]
```

```
mirror = http://ftp.us.debian.org/debian/
```

```
[...]
```

⁷ http://es.wikipedia.org/wiki/Logical_Volume_Manager

Configurar múltiple interfaces de red

Para configurar múltiples interfaces de red en Xen hemos de crear el archivo multi-network-bridge dentro del directorio scripts de xen:

```
cd /etc/xen/scripts/ && vi multi-network-bridge
```

En el archivo copiamos el código siguiente, que ejecuta varias veces el script de creación de interfaces virtuales de Xen.

```
#!/bin/sh
dir=$(dirname "$0")
"$dir/network-bridge" "$@" vifnum=0 netdev=eth0 bridge=eth0
"$dir/network-bridge" "$@" vifnum=1 netdev=eth1 bridge=eth1
```

Hay que cambiar los números de interfaz Ethernet por los correctos en caso de que sea necesario y si queremos añadir más bridges solo es necesario copiar la última línea y modificar los números de Ethernet.

Si no queremos que el padre use una IP en el segundo bridge (o en otros) editamos el archivo `/etc/network/interfaces` y añadimos:

```
iface eth1 inet manual
pre-up ifconfig $IFACE up
post-down ifconfig $IFACE down
```

Le damos permisos de ejecución:

```
chmod +x multi-network-bridge
```

Por último editamos el archivo `/etc/xen/xend-config.sxp` y cambiamos la línea (`network-script network-bridge`) por

```
(network-script multi-network-bridge)
```

Reiniciamos el ordenador y automáticamente se crearan las interfaces necesarias para virtualizar varias redes.

8. Implementación y configuración del sistema de gestión de MVs

Instalación del entorno web de VirtManager

Debido a que VirtManager esta escrito en JAVA necesitamos un servidor de aplicaciones web, en este caso nos hemos decantado por Apache Tomcat, un servidor de aplicaciones web open-source:

Primero hemos de instalar la última versión de Java Runtime mediante el gestor de paquetes apt:

```
apt-get install sun-java6-jdk
```

A continuación descargar la última versión de Apache Tomcat de su página oficial:

```
cd /usr/local wget
```

```
http://archive.apache.org/dist/tomcat/tomcat-7/v7.0.39/bin/apache-tomcat-7.0.39.tar.gz
```

```
tar -zxvf apache-tomcat-7.0.39.tar.gz
```

```
ln -s apache-tomcat-7.0.39 tomcat
```

Creamos un usuario y grupo de sistema con el cual se ejecutará Tomcat:

```
groupadd tomcat
```

```
useradd -g tomcat -d /usr/local/tomcat tomcat usermod -G www-data tomcat
```

Seguidamente creamos el fichero de arranque del init.d para Tomcat:

```
vi /etc/init.d/tomcat
```

Copiando dentro de este fichero:

```
#Tomcat auto-start
#description: Auto-starts tomcat
#processname: tomcat
#pidfile: /var/run/tomcat.pid
#this path should point to your JAVA_HOME Directory
export JAVA_HOME=/usr/lib/jvm/java-6-sun
case $1 in
start)
sh /usr/local/tomcat/bin/startup.sh ;;
stop)
sh /usr/local/tomcat/bin/shutdown.sh ;;
restart)
sh /usr/local/tomcat/bin/shutdown.sh
sh /usr/local/tomcat/bin/startup.sh ;;
```

```
esac
exit 0
```

Damos permisos de ejecución al fichero creado anteriormente:

```
chmod 755 /etc/init.d/tomcat
```

Por último, habilitamos que el Tomcat arranque cada vez que se inicie el sistema mediante el fichero dentro del directorio init.d, el cual hemos creado anteriormente:

```
update-rc.d tomcat defaults
```

A continuación instalaremos la aplicación MySQL que será el servidor donde se alojará la base de datos del sistema. Mediante apt instalamos la última versión de MySQL server y los elementos necesarios para que funcione:

```
apt-get update apt-get install mysql-server mysql-client libmysqlclient15-dev
```

IMPORTANTE: durante la instalación se pedirá la contraseña del usuario root, la cual usaremos a continuación para crear y acceder a la base de datos.

Finalmente, instalaremos configurar la aplicación VirtManager.

Primero descargamos y descomprimos los ficheros de instalación de VirtManager que están alojados en i2CAT:

```
wget http://calisto.i2cat.net/VirtManager.tar.gz
tar -zxvf VirtManager.tar.gz
```

Movemos el fichero war al directorio de aplicaciones de Tomcat:

```
mv VirtManager.war /usr/local/tomcat/webapps/
```

Creamos la base de datos de la aplicación y le cambiamos los permisos de acceso para poder acceder desde cualquier máquina:

```
mysql -u root -p
create database VirtManager;
grant all privileges on VirtManager.* to root@'%' identified by
'mypasswdroot';
exit
```

A continuación, modificamos el fichero de configuración VirtManager.properties, en la ruta

/usr/local/tomcat/webapps/VirtManager/WEBINF/classes/net/i2cat/xen/utils/conf/ dentro de este fichero asignamos la dirección IP del servidor web y los cambios de configuración necesarios para el correcto funcionamiento de VirtManager.

```
tomcat.dir=/usr/local/tomcat/
images.dir=/images/
xen.dir= /opt/ofelia/oxa/cache/vms/1/xx
server.ip=localhost
server.port=9229
webserver.ip=localhost
```



```
webserver.port=9000
```

Por último ya solo necesitamos arrancar el servicio web mediante:

```
service tomcat start
```

Y podemos acceder al servicio web VirtManager, mediante la URL `http://ip_servidor:8080/VirtManager`

Instalación del agente de gestión en las máquinas anfitrión

El agente es el encargado de comunicarse con el servidor central de VirtManager para recibir las ordenes referentes a las máquinas virtuales y actualizar el estado de las mismas.

Para instalar el agente en cada anfitrión primero ejecutaremos un script hemos creado con el fin de auto-configurar el entorno de forma rápida y sencilla para el correcto funcionamiento del agente. El script instala el software necesario para que el agente pueda funcionar (mysql-client, pymysql ...) además de generar las rutas necesarias para la gestión de las VMs por parte del agente.

Para ellos primero descargamos y ejecutamos el script:

```
wget http://calisto.i2cat.net/script.sh
chmod 755 script.sh
./script.sh
```

Por último descargamos e instalamos el agente:

```
wget http://calisto.i2cat.net/agent.tar.gz
tar -zxvf agent.tar.gz
cd agent
chmod 755 OfeliaAgent.py
./OfeliaAgent.py -b
```

Una vez hecho esto el anfitrión esta preparado para que sea controlado mediante el sistema de gestión de MVs VirtManager.

9. Resultados

El proyecto se ha implantado correctamente en dos servidores, los cuales se utilizan para hacer pruebas de funcionamiento, de funcionalidad y uso del sistema.

El resultado a sido satisfactorio ya que gracias al sistema nos es posible crear máquinas virtuales mediante una interfaz de usuario web, simple, intuitiva y apta de usar para todos los usuarios.

Además al disponer de una autenticación centralizada basada en LDAP nos es posible aplicar diferentes permisos a cada uno de los usuarios, de forma que puedan existir MVs asignadas tanto a usuarios individuales como a departamentos de la empresa, de forma que todos los miembros de ese departamento tengan acceso a dichas maquinas.

Las pruebas de rendimiento han sido satisfactorias ya que hemos observado que los servidor Host pueden absorber nuevas máquinas virtuales sin que se vean afectadas el resto.

Por otra parte, a causa de la falta de tiempo, no nos ha sido posible implementar el acceso remoto automatizado a las máquinas virtuales mediante la interfaz de usuario. Esta funcionalidad la contemplaremos en versiones futuras.

VirtManager: Interfaz de usuario

La interfaz de usuario del VirtManager esta compuesta por las siguientes secciones:

Pantalla de Inicio: El usuario se autentifica en el sistema con su usuario de LDAP de la fundación i2CAT y podrá acceder a sus MVs.

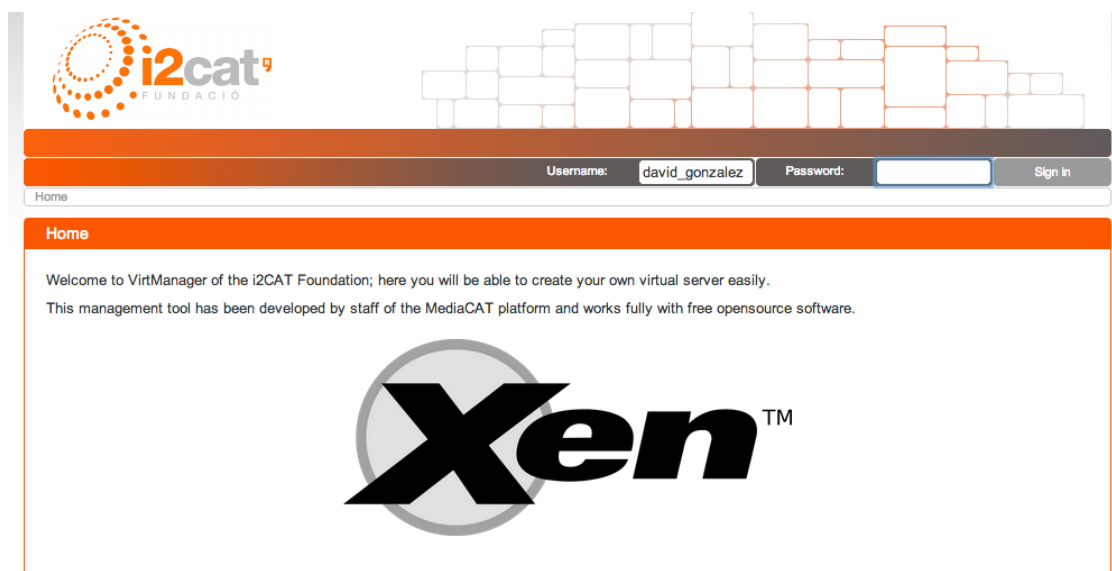


Imagen 7: Pantalla de inicio y Log In al sistema

Formulario de creación de una nueva MV: en este formulario el usuario puede elegir las características de la máquina, tales como memoria RAM, tamaños del disco duro, etc:

Create Virtual Machine

In this section, you will be able to create your own virtual machine.

Name of the machine:

Virtualization Type:

Operative System:

Host:

Disk size: (in GB)

Memory size: (in MB)

Root Password:

Assigned to:

- David Gonzalez
- David Rincón
- David Roldan
- Daniel Quintas
- David Gardella
- David Ortega
- Dave Wilson
- David O'Callaghan
- David Griffin
- Raul Landa

2013 (c) MediaCAT

Imagen 8: Formulario de creación de MV

Lista de MVs del usuario: el usuario tiene acceso a sus Mvs, permitiendo le apagar, encender, reiniciar y eliminar, además de consultar los datos de las mismas.

Virtual Machines List

Hostname	Disk size	Memory size	Swap size	Operative System	Node	Assigned To	Extra Info	Actions
prueba1.i2cat.net	10 GB	512 MB	1024 MB	debian-squeeze-xen4-rarp	chucknorris		SHOW	STOP, RESTART, DELETE, EDIT
prueba10.i2cat.net	10 GB	512 MB	1024 MB	debian-squeeze-xen4-rarp	chucknorris		SHOW	STOP, RESTART, DELETE, EDIT
prueba11.i2cat.net	10 GB	512 MB	1024 MB	debian-squeeze-xen4-rarp	chucknorris		SHOW	STOP, RESTART, DELETE, EDIT
prueba12.i2cat.net	10 GB	512 MB	1024	debian-squeeze-xen4-rarp	chucknorris		SHOW	STOP, RESTART, DELETE, EDIT
prueba13.i2cat.net	10 GB	512 MB	1024	debian-squeeze-xen4-rarp	chucknorris		SHOW	STOP, RESTART, DELETE, EDIT
prueba2.i2cat.net	10 GB	512 MB	1024	debian-squeeze-xen4-rarp	chucknorris		SHOW	STOP, RESTART, DELETE, EDIT
prueba3.i2cat.net	10 GB	512 MB	1024	debian-squeeze-xen4-rarp	chucknorris		SHOW	STOP, RESTART, DELETE, EDIT
prueba4.i2cat.net	10 GB	512 MB	1024	debian-squeeze-xen4-rarp	chucknorris		SHOW	STOP, RESTART, DELETE, EDIT
prueba5.i2cat.net	10 GB	512 MB	1024	debian-squeeze-xen4-rarp	chucknorris		SHOW	STOP, RESTART, DELETE, EDIT
prueba6.i2cat.net	10 GB	512 MB	1024	debian-squeeze-xen4-rarp	chucknorris		SHOW	STOP, RESTART, DELETE, EDIT

Extra Information

This information may be outdated with the server

Root Password: [REDACTED]

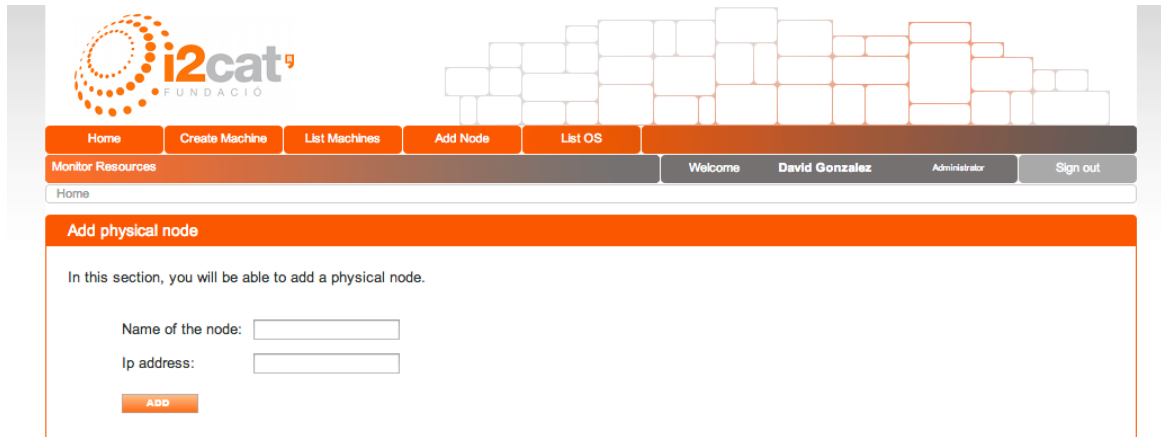
IP Address: [REDACTED]

MAC Address: 00:16:3E:20:A1:75

First Prev 12 Next Last

Imagen 9: Lista de máquinas virtuales

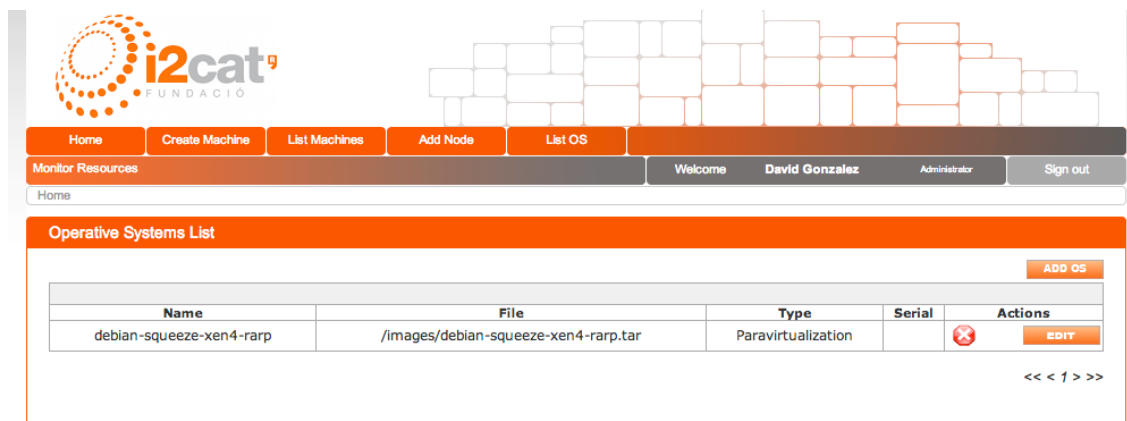
Formulario de configuración de un nuevo servidor Padre: Una vez correctamente configurado un servidor Padre se añade al sistema mediante este formulario, asignándole un nombre y poniendo su IP de forma que se puedan crear nuevas máquinas virtuales en este servidor Padre.





The screenshot shows the i2cat web interface. At the top left is the i2cat logo. Below it is a navigation menu with buttons for Home, Create Machine, List Machines, Add Node, and List OS. A secondary menu shows 'Monitor Resources' and a user profile for 'David Gonzalez' with 'Welcome', 'Administrator', and 'Sign out' options. The main content area is titled 'Add physical node' and contains the text: 'In this section, you will be able to add a physical node.' Below this text are two input fields: 'Name of the node:' and 'Ip address:'. At the bottom of the form is an 'ADD' button.

Imagen 10: Formulario de configuración de un nuevo servidor Padre

Lista de sistemas operativos disponibles: En esta pantalla están listados todos los sistemas operativos disponibles para el usuario, además los administradores pueden añadir nuevos sistemas operativos mediante el botón ADD OS.



The screenshot shows the i2cat web interface with the 'Operative Systems List' section. At the top right of this section is an 'ADD OS' button. Below it is a table with the following data:

Name	File	Type	Serial	Actions
debian-squeeze-xen4-rarp	/images/debian-squeeze-xen4-rarp.tar	Paravirtualization		 

At the bottom right of the table area are navigation arrows: '<< < > >>'.

Imagen 11: Lista de sistemas operativos disponibles

Pantalla de monitorización de los recursos del sistema: En esta pantalla se puede monitorizar el estado de cada uno de los servidores padre:

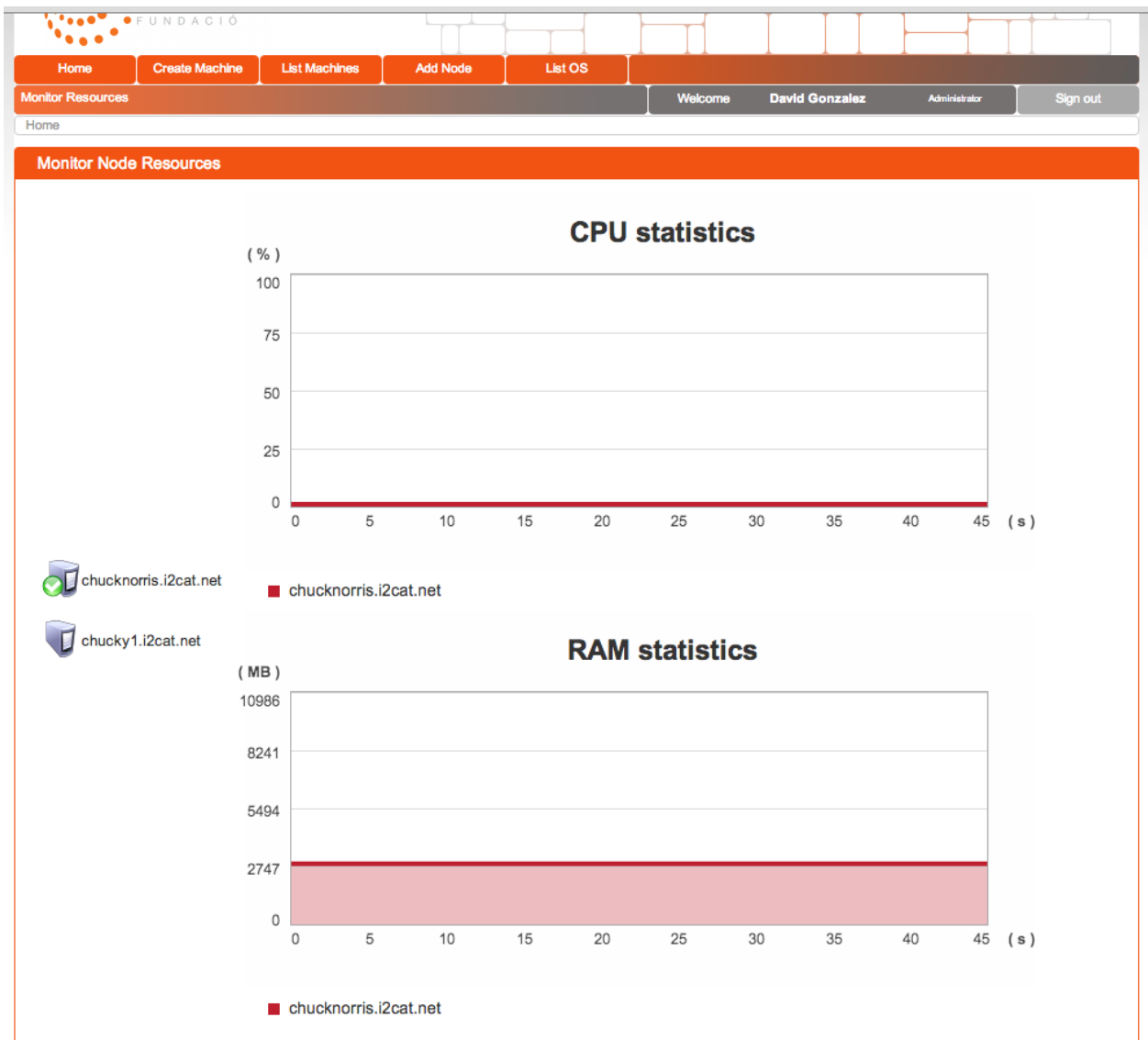


Imagen 12: Pantalla de monitorización de los recursos del sistema

10. Valoración económica

Planificación de recursos de personal

Siguiendo la planificación inicial se ha dividido el proyecto entre las diferentes tareas siguiendo la siguiente forma:

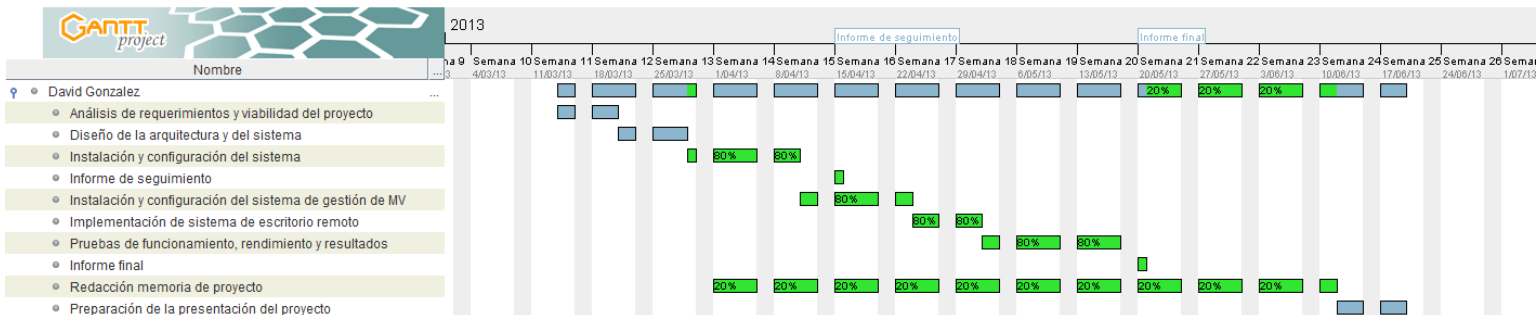


Imagen 13: Gantt de recursos de personal

Según la planificación el proyecto podemos calcular el número de horas totales partiendo de una aproximación de 4 horas de trabajo dedicadas por día laborable, durante la realización del proyecto. A partir de el número de horas y teniendo en cuenta que mi coste por hora es de 24€/hora, la valoración económica de cada una de las tareas ha sido:

	Horas	Dedicación	Horas reales	Total
Análisis de requerimientos y viabilidad del proyecto	20	100%	20	480 €
Diseño de la arquitectura y del sistema	24	100%	24	576 €
Instalación y configuración del sistema	36	80%	28,8	691,2 €
Instalación y configuración del sistema de gestión	36	80%	28,8	691,2 €
Implementación de sistema de escritorio remoto	24	80%	19,2	460,8 €
Pruebas de funcionamiento, rendimiento y resultados	48	80%	38,4	921,6 €
Redacción memoria de proyecto	208	20%	41,6	998,4 €
Preparación de la presentación del proyecto	24	100%	24	576 €
Totales			224,8	5395,2 €

Si solo tenemos en cuenta las horas dedicadas dentro del horario de trabajo, la valorización económica del coste del proyecto para la empresa ha sido de 5395,2 €, si tenemos en cuenta que el precio aproximado de un servidor es de 3000 €, si con este proyecto podemos ahorrar la compra de 2 servidores ya podríamos dar por rentabilidad la inversión de horas para la empresa.

De hecho la rentabilidad es mayor ya que en este caso no hemos tenido en cuenta el consumo eléctrico que generan estos servidores:

Impacto medioambiental y económico

El continuo crecimiento de aplicaciones y servicios disponibles en la sociedad actual implica el incremento de la capacidad tanto de infraestructura como de consumo energético de los servidores que ofrecen estos servicios y aplicaciones. Debido a este crecimiento el concepto de computación en la nube se ha convertido en un elemento clave dentro de los centros de datos de hoy en día, ya que ayuda a optimizar los recursos del sistema y por ende a reducir el coste económico para las empresas que los administran.

El hecho que el consumo energético representa un gran problema en los centros de datos se ve reflejado en un informe realizado por la Agencia de Protección Medioambiental americana en el que se estimó que entre el 2000 y 2006 el consumo energético se había duplicado. A pesar de ello en un informe realizado por el departamento de ingeniería medioambiental de la Universidad de Stanford, sobre el consumo energético de los centros de datos entre 2005 y 2010 afirma que a pesar que existe un incremento significativo se observa una tendencia a contraerse.

Entre las posibles causas que se exponen en este informe afirman que se debe a la aparición de tecnologías más eficientes energéticamente y a la implantación cada vez más común de virtualización de servidores. Precisamente la virtualización que emplea la computación en la nube para proporcionar servicios, representa el principal método para reducir este consumo energético.

Por estas razones, y a pesar de que no hemos podido realizar medidas que demuestren esta teoría, podemos decir que la implantación de este proyecto puede suponer un gran ahorro energético para la empresa y por ende un impacto medioambiental menor.

11. Conclusiones

Este proyecto me ha permitido profundizar en un campo de la administración de sistemas que me apasiona, la virtualización de servidores y la computación en la nube, lo que me ha permitido aprender mucho y fomentar mi carrera profesional en este aspecto.

El proyecto ha sido un éxito, a pesar de que el tiempo era muy ajustado y este proyecto se tendría que haber planteado con una planificación más espaciada en el tiempo, hemos visto que utilizando tecnologías y software libre podemos desarrollar una nube de computación privada que cumpla con las expectativas de las empresas para alojar en ella, tanto sus servicios como los entornos de escritorios de sus usuarios.

Trabajos futuros

A corto plazo:

Como ya hemos comentado en el punto anterior Resultados, no ha sido posible realizar la implementación del acceso remoto automatizado a las máquinas virtuales mediante la interfaz de usuario, a pesar que ya ha habido un trabajo previo de investigación de las tecnologías disponibles y hemos visto que es factible hacerlo mediante FreeNX. Por esta razón dentro de nuestra planificación a corto plazo se haya la implementación de esta nueva funcionalidad.

A medio/largo plazo:

Sería interesante agregar una nueva funcionalidad que permita la migración de MVs entre diferentes servidores padre de forma que se puedan liberar los recursos de un servidor sin tener que afectar al servicio de las MVs.

Por otra parte también sería muy interesante agregar una capa de inteligencia artificial entre la interfaz de usuario y los servidores padre, de forma que el sistema sea capaz de responder ante errores de hardware o incluso realizar un mejor aprovechamiento de los recursos del sistema.

Por último una nueva funcionalidad importante sería soportar otros software e virtualización del mercado, tales como VMware o KVM, sin que se viera afectado el funcionamiento del sistema.