



ESTUDIO DE RENDIMIENTO Y FUNCIONALIDAD  
SOBRE DIFERENTES SOLUCIONES DE  
VIRTUALIZACIÓN

Memoria del proyecto de final de carrera correspondiente a los estudios de Ingeniería Superior en Informática presentado por Alberto Garcia Calahorro y dirigido por Josep Prieto Blázquez.

Bellaterra, Junio de 2009

El firmante, Josep Prieto Blázquez, profesor del Departament d'Enginyeria de la Informació i de les Comunicacions de la Universitat Autònoma de Barcelona

CERTIFICA:

Que la presente memoria ha sido realizada bajo su dirección por Alberto Garcia Calahorro

Bellaterra, Junio de 2009

---

Firmado: Josep Prieto Blázquez

*Educación no es dar carrera para vivir,  
sino templar el alma para las dificultades de la vida.*  
*Pitágoras.*



# Índice general

<b>1. Introducción</b>	<b>1</b>
1.1. Motivación y justificación del proyecto . . . . .	1
1.2. Descripción general del trabajo realizado . . . . .	2
1.3. Objetivos del proyecto . . . . .	3
1.4. Estructura de la memoria . . . . .	4
<b>2. Estado del arte</b>	<b>5</b>
2.1. Definición de virtualización . . . . .	5
2.1.1. Usos de la virtualización . . . . .	7
2.2. Tipos de máquinas virtuales . . . . .	8
2.2.1. Máquinas virtuales de sistema . . . . .	8
2.2.2. Máquinas virtuales de proceso . . . . .	9
2.3. Técnicas de virtualización . . . . .	9
2.3.1. Emulación . . . . .	10
2.3.2. Virtualización parcial . . . . .	10
2.3.3. Virtualización completa . . . . .	11
2.3.4. Paravirtualización . . . . .	11
2.4. Soluciones existentes en el mercado . . . . .	12
2.4.1. VMware . . . . .	13
2.4.2. Microsoft . . . . .	13
2.4.3. VirtualBox . . . . .	13
2.4.4. XEN . . . . .	14
2.4.5. Qemu . . . . .	15

2.5. Comparativas . . . . .	16
2.6. Conclusiones . . . . .	18
<b>3. Estudio de rendimiento</b>	<b>19</b>
3.1. Red . . . . .	20
3.1.1. Primer juego de pruebas . . . . .	20
3.1.2. Segundo juego de pruebas . . . . .	28
3.2. CPU . . . . .	36
3.2.1. Resultado de las pruebas de factorización . . . . .	37
3.3. Memoria . . . . .	38
3.4. Conclusiones . . . . .	40
<b>4. Diseño y desarrollo de la aplicación</b>	<b>43</b>
4.1. Especificación de requisitos . . . . .	43
4.1.1. Software de virtualización . . . . .	43
4.1.2. Software adicional . . . . .	45
4.1.3. Hardware . . . . .	46
4.2. Diseño de la aplicación . . . . .	46
4.2.1. Creación de la máquina virtual . . . . .	47
4.2.2. Control de encendido y apagado de la máquina virtual . . . . .	48
4.2.3. Modificación de las máquinas virtuales . . . . .	49
4.2.4. Estado de las máquinas virtuales . . . . .	49
4.3. Características de la aplicación . . . . .	51
4.4. Acceso a la aplicación . . . . .	52
4.5. Posibles vías de desarrollo . . . . .	53
4.6. Conclusiones . . . . .	54
<b>5. Conclusiones</b>	<b>57</b>
<b>Bibliografía</b>	<b>59</b>

# Índice de figuras

2.1. Virtualización . . . . .	6
2.2. Emulación . . . . .	10
2.3. Virtualización completa . . . . .	11
2.4. Resultado búsqueda VM's de software libre . . . . .	17
2.5. Resultado búsqueda VM's de software privativo . . . . .	17
3.1. Resultado primer juego de pruebas p2p. . . . .	21
3.2. Gráfica de los resultados anteriores. . . . .	28
3.3. Resumen del segundo juego de prueba p2p. . . . .	29
3.4. Gráfica de los resultados del segundo juego de pruebas . . . . .	35
3.5. Comparativa de los tiempos obtenidos . . . . .	38
3.6. Tiempos de renderizado de VMware utilizando varias VM's. . . . .	40
4.1. Resumen aplicación. . . . .	47
4.2. Diagrama de flujo de la creación de VM. . . . .	48
4.3. Diagrama de flujo de Encendido/Apagado de las VM. . . . .	49
4.4. Diagrama de la sección de modificación de las máquinas virtuales. . . . .	50
4.5. Diagrama del apartado de estado de las máquinas virtuales. . . . .	50
4.6. Detalle de la sección de creación de máquinas virtuales. . . . .	52
4.7. Máquina virtual utilizando rdesktop. . . . .	53

# Capítulo 1

## Introducción

En este primer capítulo haremos una pequeña introducción al mundo de la virtualización. En capítulos posteriores ampliaremos el concepto de virtualización, y daremos a conocer las motivaciones que nos han llevado a la elaboración de este proyecto.

### 1.1. Motivación y justificación del proyecto

El uso de máquinas virtuales se ha convertido en una tecnología bastante frecuente, ya no solo en el ámbito empresarial, sino también entre los usuarios domésticos, siendo una de las tecnologías con mayor crecimiento en los últimos tiempos. Con estos programas, es posible disponer de varios sistemas operativos ejecutándose simultáneamente y de forma totalmente aislada. De esta manera, podemos ejecutar aplicaciones, formatear o reinstalar otro sistema operativo en nuestro disco duro virtual, sin necesidad de reiniciar el sistema anfitrión<sup>1</sup>.

Una máquina virtual es un software que emula a un ordenador y puede ejecutar programas como si fuese un ordenador real [1]. Esto significa que podemos emular un ordenador dentro de otro, lo que nos ofrece ventajas inmediatas: podemos instalar y ejecutar varios sistemas operativos a la vez, cada uno de ellos corriendo bajo una máquina emulada sobre una única máquina anfitrión, reduc-

---

<sup>1</sup>Llamamos sistema anfitrión al sistema operativo instalado en nuestro computador.

ción de costes de mantenimiento (consumo eléctrico, reducción de los contratos de mantenimiento de maquinaria), etc.

Teniendo esto presente y mirando el panorama actual en que se encuentran los computadores de hoy en día, donde ya disponemos de procesadores de varios núcleos y gran cantidad de memoria, la cuestión que se nos plantea es si con la utilización de las máquinas virtuales podemos mejorar el rendimiento de estas computadoras en apartados como el procesador, la memoria, o el tráfico de red. O bien, utilizarlas para obtener un mayor rendimiento de la máquina anfitriona, a partir del uso concurrente que los usuarios puedan realizar. Esto último lo realizaremos con el desarrollo de nuestra aplicación.

Hay que tener presente que el rendimiento de las máquinas virtuales puede variar dependiendo del método de virtualización utilizado en cada caso. Como veremos en capítulos posteriores, dependiendo de qué técnica de virtualización se utilice, se puede obtener diferentes rangos de prestaciones.

También cabe destacar el momento actual en que se encuentran la mayoría de sistemas operativos. De unos años atrás hasta la fecha, se ha pasado de tener un sistema operativo con el que teníamos que convivir durante largos periodos de tiempo (con sus necesarios formateos para volver a un rendimiento óptimo), a operativos mucho más estables y fiables con continuas actualizaciones que hacen que no necesitemos reinstalarlos con tanta frecuencia. Esto nos puede llevar a dejar de lado la utilización o pruebas de otro tipo de sistemas operativos, pero gracias a la virtualización, manteniendo nuestro sistema anfitrión intacto, podemos tener varias máquinas virtuales con sistemas operativos diferentes.

## 1.2. Descripción general del trabajo realizado

Durante el transcurso de este proyecto se han marcado unos puntos de estudio en los que centrarse, para conocer, si la virtualización nos ofrece algún tipo de mejora como se comentaba anteriormente. Los puntos de estudio en los que nos centraremos serán los siguientes:

- Red: mediante la utilización de un software de transferencia de datos, es-

tudiaremos el rendimiento que obtenemos de la conexión de red al utilizar varias máquinas virtuales, frente a la que podemos obtener con la utilización de la máquina nativa.

- Procesador: la idea de esta parte es someter al procesador a una prueba de stress<sup>2</sup> con un programa que se centre en el uso del procesador. La herramienta elegida es el programa msieve<sup>3</sup> [2], con el que factorizaremos un número  $n$  para encontrar los dos primeros que lo componen, siguiendo el patrón del RSA<sup>4</sup>.

$$n = p * q$$

- Memoria: usaremos un programa gráfico, en este caso Blender, para renderizar unas imágenes, cosa que a priori supondrá un gasto importante de memoria y veremos como se comportan los sistemas virtualizados frente a su uso normal en sistema anfitrión.
- Gestión de usuarios: el hecho que en una máquina anfitriona puedan ejecutarse diversas sistemas virtualizados, nos plantea la cuestión de como se podría aprovechar esta funcionalidad de la virtualización centrada en la gestión de los usuarios y en los usos que estos le dan a las máquinas. Esto lo veremos con la creación de nuestra aplicación, a partir de la cual permitiremos la creación de máquinas virtuales, con ciertos parámetros configurables.

### 1.3. Objetivos del proyecto

El objetivo principal de este proyecto es comprobar si con el uso de la virtualización podemos obtener mejoras respecto a sistemas sin virtualizar, aprove-

---

<sup>2</sup>Este tipo de prueba se realiza para determinar la solidez del sistema en momentos de carga extrema y ayuda a los administradores para determinar si la aplicación rendirá lo suficiente en caso de que la carga real supere a la carga esperada.

<sup>3</sup>Msieve es una librería y aplicación para factorizar números grandes, utilizando potentes algoritmos.

<sup>4</sup>El sistema criptográfico con clave pública RSA es un algoritmo asimétrico cifrador de bloques, que utiliza una clave pública, la cual se distribuye (en forma autenticada preferentemente), y otra privada, la cual es guardada en secreto por su propietario.

chando los avances tecnológicos actuales. Para ello, realizaremos un estudio de rendimiento centrado en tres componentes principales de nuestro ordenador, estos son: la conexión de red, el uso del procesador y la memoria. Además, abordaremos un cuarto aspecto referido a la gestión multiusuario que se puede realizar con la utilización de las máquinas virtuales.

Otro de los objetivos del proyecto consiste en el diseño e implementación de una aplicación que aprovechará los datos obtenidos en el estudio anteriormente citado y se realizará con el software de virtualización más conveniente. Para ello, será necesario realizar un estudio de la disponibilidad y facilidades de uso que nos ofrezca cada herramienta de virtualización.

## **1.4. Estructura de la memoria**

La estructura de la memoria es la siguiente:

- En el segundo capítulo, titulado estado del arte, se dará una definición de lo que es una máquina virtual y veremos que tipos y técnicas de virtualización tenemos disponibles, así como la descripción de algunos ejemplos de aplicaciones de virtualización más conocidas.
- En el tercer capítulo se verán las tres áreas de estudio en las que se centra este proyecto, así como la metodología y los resultados obtenidos en sus pruebas pertinentes.
- En el capítulo 4 se presentará el diseño y funcionamiento de la aplicación, capaz de gestionar diferentes máquinas virtuales adaptadas a las necesidades de los usuarios.
- Y para finalizar, el último capítulo describirá las conclusiones obtenidas con las pruebas realizadas durante el transcurso del PFC.

# Capítulo 2

## Estado del arte

En este segundo capítulo definiremos el concepto de virtualización, mostraremos las diferentes técnicas que existen a la hora de virtualizar y daremos algunos ejemplos de aplicaciones existentes que realizan este proceso. Para finalizar el capítulo, veremos unas comparativas de soluciones de virtualización existentes, que nos ayudará a decidir cual de ellas utilizaremos en el posterior desarrollo.

### 2.1. Definición de virtualización

La virtualización es una técnica que permite ocultar, a través de la encapsulación, las características físicas de los recursos informáticos cuando otros sistemas, aplicaciones o usuarios finales interactúan con estos recursos [3]. Las máquinas virtuales (*Virtual Machine*, VM) fueron definidas originalmente, en 1974, por Popek y Golberg como «*an efficient, isolated duplicate of a real machine*»[7], aunque actualmente esta definición incluye máquinas virtuales que no tienen una correspondencia con hardware real, como pudiera ser la *Java Virtual Machine*.

IBM fue quien empezó a implementar la virtualización en la década de los sesenta, como una manera lógica de particionar ordenadores *mainframe*<sup>1</sup> en máquinas virtuales independientes [4]. Estas particiones permitían a estas computadoras,

---

<sup>1</sup>Un mainframe o computadora central es una computadora grande, potente y costosa usada principalmente por una gran compañía para el procesamiento de una gran cantidad de datos; por ejemplo, para el procesamiento de transacciones bancarias.

realizar varias tareas al mismo tiempo. Dado que en aquella época los *mainframes* eran recursos caros, se diseñaron para ser particionados para aprovechar de esta manera, al máximo, la inversión [5].

La virtualización fue dejada de lado durante los ochenta y noventa, cuando las aplicaciones cliente-servidor y los equipos económicos establecieron el modelo de informática distribuida. El incremento de implementaciones de servidores y escritorios x86<sup>2</sup> generó nuevos problemas operacionales y de infraestructura como pueden ser: baja utilización de la infraestructura (las aplicaciones típicas de servidores x86 logran una utilización media de entre un 10 % y un 15 %), incremento de los costes en infraestructura física, etc.

A finales de los años noventa, principios del 2000, empezó a resurgir esta tecnología, incluso para usuarios domésticos, incrementando su uso y popularidad hasta la fecha. Con la aparición de nuevas tecnologías en el campo del hardware por parte de Intel<sup>3</sup> y AMD<sup>4</sup>, se logra mejorar el rendimiento de las máquinas virtuales.

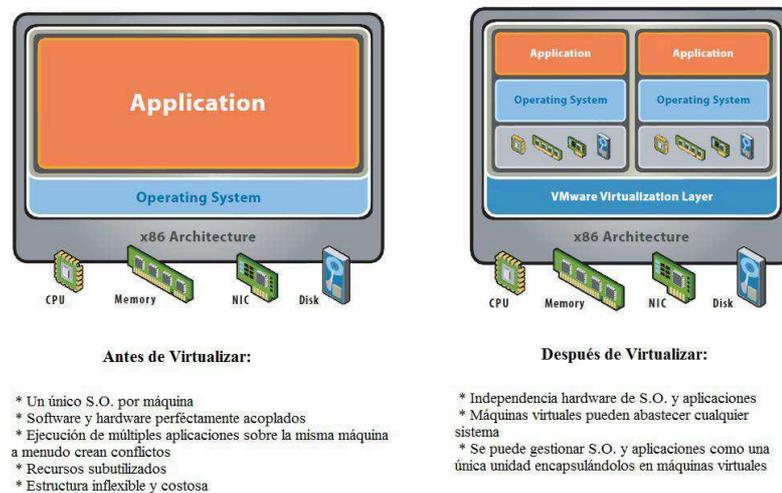


Figura 2.1: Virtualización

<sup>2</sup>x86 es la denominación genérica dada a ciertos microprocesadores de la familia Intel, sus compatibles y la arquitectura básica a la que estos procesadores pertenecen.

<sup>3</sup>Intel VT: la tecnología de virtualización Intel asistida por hardware.

<sup>4</sup>AMD-V: AMD Virtualization.

El *hipervisor*, o *Virtual Machine Monitor* (VMM), es el responsable de manejar los recursos de la máquina física (CPU, memoria, red y almacenamiento), de forma que el usuario pueda crear diversas máquinas virtuales, presentando a cada una de ellas, una interfaz del hardware que sea compatible con el sistema operativo elegido. Esta capa de software, gestiona los recursos principales del ordenador (memoria, procesador, disco y red) para repartir dinámicamente la carga entre los diferentes sistemas virtualizados. Podemos distinguir dos tipos de hipervisor:

1. Hipervisor tipo 1: también denominado nativo o *unhosted*, se ejecuta directamente sobre el hardware, para ofrecer la funcionalidad descrita. Ejemplos actuales que utilizan este tipo de hipervisor serían: Oracle VM, Vmware's ESX Server o XEN.
2. Hipervisor tipo 2: también denominado *hosted*, se ejecuta sobre un sistema operativo. Ejemplos de este tipo son: Vmware Server, Vmware Workstation, Qemu, Microsoft Virtual PC, VirtualBox, entre otros.

### 2.1.1. Usos de la virtualización

Algunas de las funcionalidades que la virtualización nos puede ofrecer son las siguientes [6] :

- Consolidación de servidores y optimización de infraestructuras: la virtualización permite aumentar la utilización de los recursos, mediante la agrupación de recursos de infraestructura comunes y la superación del modelo 'una aplicación para un servidor'.
- Reducción de costes en infraestructura física: con la virtualización se puede reducir el número de servidores y hardware en los centros de datos. Esto comporta una disminución de los requisitos inmobiliarios, de consumo eléctrico, refrigeración, etc.
- Flexibilidad y capacidad de respuesta: la virtualización permite a los administradores dedicar menos tiempo a tareas repetitivas como mantenimiento, configuración y supervisión.

- Mayor disponibilidad: eliminación de paradas planificadas, recuperación rápida frente a paradas no planificadas como cortes de suministro eléctrico.
- Capacidad de gestión y seguridad: se puede implementar, administrar y supervisar entornos de escritorio protegido a los que los usuarios pueden acceder localmente o de forma remota, desde cualquier ordenador.

## 2.2. Tipos de máquinas virtuales

Las VM las podemos clasificar en dos categorías, descritas a continuación:

- Máquinas virtuales de sistema (en inglés *System Virtual Machine*).
- Máquinas virtuales de proceso (en inglés *Process Virtual Machine*).

### 2.2.1. Máquinas virtuales de sistema

Las máquinas virtuales de sistema, también conocidas como *hardware VM*, proporcionan un entorno completo, en el que múltiples sistemas operativos y procesos de múltiples usuarios pueden coexistir. Las máquinas virtuales de sistema surgieron en la década de los 60 y principios de los 70 y fueron el origen del término máquina virtual. Al mismo tiempo, los ordenadores de la época eran muy grandes y caros y normalmente eran compartidos por un alto número de usuarios; con las *VM's*, diferentes usuarios podrían haber ejecutado diferentes sistemas operativos sobre la misma máquina.

#### Aplicaciones de las máquinas virtuales de sistema

Diferentes sistemas operativos pueden coexistir en una misma máquina. Las máquinas virtuales pueden proporcionar una arquitectura de instrucciones (ISA) que sea diferente a la de la máquina en la que se ejecuta (host), lo que implica, que puede simular hardware. Varias máquinas virtuales, pueden ser utilizadas para la consolidación de servidores. Esto permite que servicios que normalmente se ejecutan en máquinas diferentes para evitar interferencias, se puedan ejecutar en la

misma computadora de forma completamente aislada y compartiendo los recursos de una única máquina. Esto también contribuye a reducir el coste total de las instalaciones para mantener el servicio.

### 2.2.2. Máquinas virtuales de proceso

Una máquina virtual de proceso, o máquina virtual de aplicación, se ejecuta como un proceso normal dentro de un sistema operativo, creando un pequeño entorno virtual que contiene los componentes necesarios para su ejecución. Este entorno virtual actúa como una capa entre la aplicación y el sistema operativo, y elimina los conflictos entre aplicaciones, y aplicaciones y el sistema operativo. El ejemplo más característico de este tipo de VM sería la *Java Virtual Machine* de Sun.

## 2.3. Técnicas de virtualización

Dentro del campo de la virtualización y dejando de lado las máquinas virtuales de proceso, ya que no forman parte del estudio que realizaremos, podemos encontrar diferentes técnicas o tácticas de virtualización diferentes. Estas técnicas son las siguientes:

- Emulación.
- Virtualización parcial.
- Virtualización completa.
- Paravirtualización.

Una vez enumeradas, a continuación se describirán brevemente en que consisten cada una de estas técnicas.

### 2.3.1. Emulación

Esta técnica se basa en crear máquinas virtuales que simulen el hardware de una o varias plataformas hardware diferentes. Este tipo de virtualización es la más costosa, así como la menos eficiente, ya que obliga a simular completamente el funcionamiento de una plataforma hardware e implica, también, que cada instrucción ejecutada en estas plataformas sea traducida al hardware real. En la figura 2.2, podemos ver la capa creada para simular el hardware necesario para ejecutar al sistema operativo huésped.

Pese a esto, la emulación tiene características interesantes, como poder ejecutar un sistema operativo diseñado para una plataforma específica sobre otra plataforma con la cual técnicamente no es compatible. Un representante de este tipo de virtualización es QEMU, que nos permite emular plataformas hardware como x86, PowerPC<sup>5</sup> o SPARC<sup>6</sup>.

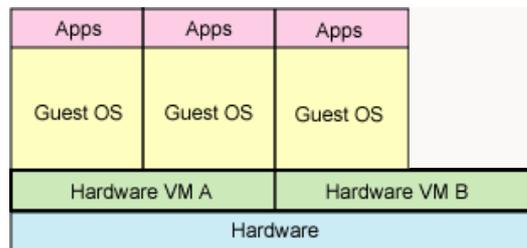


Figura 2.2: Emulación

### 2.3.2. Virtualización parcial

La máquina virtual simula múltiples instancias del entorno subyacente del hardware. Este entorno admite compartir recursos y aislar procesos, pero no permite instancias separadas de sistemas operativos *guest*. Aunque no es vista como dentro de la categoría de máquina virtual, históricamente este fue un importan-

<sup>5</sup>PowerPC (usualmente abreviada PPC) es el nombre original de la arquitectura de computadoras de tipo RISC, fue desarrollada por IBM, Motorola y Apple.

<sup>6</sup>SPARC (del inglés Scalable Processor ARChitecture) es una arquitectura RISC big-endian. Es decir, una arquitectura con un conjunto reducido de instrucciones.

te acercamiento, y fue usado en sistemas como CTSS[8] o el experimental IBM M44/44X<sup>7</sup>.

### 2.3.3. Virtualización completa

La virtualización completa (figura 2.3) nos permite ejecutar un *guest* sobre un *host* sin tener que hacer modificaciones, mediante la utilización de un *hipervisor* que permite compartir el hardware real. En cuanto al rendimiento que podemos obtener con esta técnica, decir que a priori, es superior al conseguido con la emulación. Y con las mejoras técnicas que nos ofrecen hoy en día Intel y AMD, el rendimiento que se pueden obtener, puede ser prácticamente idéntico al de la máquina nativa.

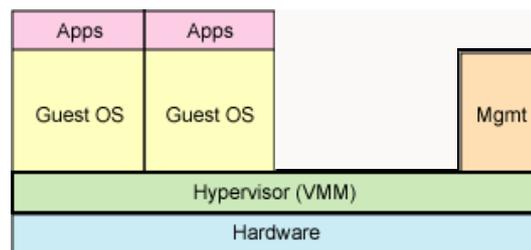


Figura 2.3: Virtualización completa

### 2.3.4. Paravirtualización

Con esta técnica se busca mejorar la eficiencia de las máquinas virtuales. Se basa en que los *guests* han de estar basados en sistemas operativos especialmente modificados para poder ejecutarse sobre un hipervisor, de manera que no sea necesario que se monitoricen todas las instrucciones, sino que, *guests* y *host* colaboren. Como representante de paravirtualización encontramos a XEN.

<sup>7</sup>Fue un computador experimental de mediados de los 60 basado en el IBM 7044, y simulaba múltiples máquinas virtuales 7044.

## 2.4. Soluciones existentes en el mercado

Para hacer un posterior estudio más concreto sobre las máquinas virtuales, lo primero que haremos será ver que soluciones existen en la actualidad en el mercado, tanto aplicaciones basadas en software libre (con licencias GPL<sup>8</sup>, LGPL<sup>9</sup>), como privativas<sup>10</sup>, esto lo podemos ver resumido en la tabla 2.1. Obviamente, debido al ámbito y la limitación de recursos a los que corresponde este proyecto, no serán elegidas aquellas aplicaciones que no nos faciliten la suficiente documentación, ni versiones, como mínimo, de prueba.

Aplicación	Tipo de licencia
VirtualBox	GPL v2
Open VZ	GPL
Bochs	LGPL
XEN	GPL
QEMU	GPL/LGPL
Virtual PC	Privativa
Virtual server	Privativa
Vmware	Privativa
Parallels	Privativa

Tabla 2.1: Listado de algunas soluciones en el campo de la virtualización.

A continuación se describen brevemente algunas de las soluciones anteriormente nombradas. Se acompaña a la descripción con una serie de ventajas y desventajas de cada solución nombrada que nos ayudará, posteriormente, a elegir a las candidatas para el estudio posterior.

<sup>8</sup>La licencia GPL (General Public License) es una licencia creada por la Free Software Foundation mediados de los 80, y está orientada principalmente a proteger la libre distribución, modificación y uso de software.

<sup>9</sup>La GNU Lesser General Public License (LGPL), se aplica en ciertas librerías, y es algo diferente a la GPL ordinaria. Esta licencia se utiliza para ciertas librerías que necesiten permitir el enlazado con programas Privativos (no Libres).

<sup>10</sup>El software propietario o privativo se refiere a cualquier programa informático en el que los usuarios tienen limitadas las posibilidades de usarlo, modificarlo o redistribuirlo, o cuyo código fuente no está disponible o el acceso a este se encuentra restringido.

### 2.4.1. VMware

Es una de las aplicaciones de virtualización más extendidas y populares. Hay fuentes [9] que indican que domina el 85 % del mercado.

VMware inserta directamente una capa de software al hardware del computador o al sistema operativo anfitrión. Esta capa crea máquinas virtuales y contiene un *hipervisor* que asigna los recursos de hardware de forma dinámica y transparente, para poder ejecutar diferentes sistemas operativos de forma simultánea en una única máquina física.

**Ventajas:** estabilidad, seguridad, admite drivers dentro del entorno emulado.

**Desventajas:** el gestor de máquinas virtuales, no tiene un buen rendimiento en máquinas con hardware antiguo; código privativo.

**Coste:** VMware dispone de varias aplicaciones en el mercado (desde unos 100 dólares a más de 1000), alguna de ellas de forma gratuita (VMware player) o con versiones de prueba.

### 2.4.2. Microsoft

Dispone de dos productos en el campo de la virtualización, estos son Microsoft Virtual Pc y Microsoft Virtual Server. Virtual PC está orientada al uso personal, mientras que Virtual Server está pensada para entornos profesionales.

**Ventajas:** integración total en entornos Windows, fácil instalación y configuración.

**Desventajas:** las máquinas virtuales solo pueden ser de la familia Windows; consumo excesivo de recursos.

**Coste:** Virtual PC 2007 se encuentra disponible de forma gratuita. Virtual Server es gratuito.

### 2.4.3. VirtualBox

Esta es una de la máquinas virtuales libres disponible en el mercado, aunque existen versiones comerciales. Fue inicialmente desarrollado por la empresa *Innotek*, pero en 2008 fue adquirido por *Sun Microsystems*. Inicialmente tuvo licencia

de software privativo, pero en 2007 apareció VirtualBox OSE (Open Source Edition), con licencia GPL 2.

Comparándola con otras aplicaciones privadas de virtualización, como VMware workstation o Microsoft Virtual PC, VirtualBox carece de algunas funcionalidades (aunque últimamente está recibiendo numerosas actualizaciones), pero provee de otras como la ejecución de máquinas virtuales de forma remota, por medio del Remote Desktop Protocol (RDP<sup>11</sup>). Se distribuye con dos modalidades de licencia:

1. Una versión con licencia GPL.
2. Una versión con licencia PUEL<sup>12</sup>.

**Ventajas:** excelente documentación; aplicación de código abierto; acceso a la consola de cada máquina virtual por red vía *Remote Display Protocol* (RDP); fácil instalación.

**Desventajas:** no dispone de aceleración 3D, ya que solo se tiene acceso a la tarjeta gráfica virtual de VirtualBox.

**Coste:** gratuito.

#### 2.4.4. XEN

XEN es una VM desarrollada por la Universidad de Cambridge bajo licencia GPL. En 2007 Citrix adquirió XenSource, y actualmente podemos encontrar dos productos, uno de uso libre y otro privativo orientado a uso profesional.

Utiliza la técnica de la paravirtualización para alcanzar un alto rendimiento. A diferencia de otros productos de virtualización que funciona sobre el sistema operativo, Xen se ejecuta directamente sobre el hardware. Cuando se inicia un sistema virtualizado con Xen, es este el que efectúa el *boot* de la máquina y el que posteriormente inicia las máquinas virtuales. La máquina virtual inicial se llama

---

<sup>11</sup>Protocolo desarrollado por Microsoft que permite la comunicación en la ejecución de una aplicación entre un terminal (mostrando la información procesada que recibe del servidor) y un servidor (recibiendo la información ingresada por el usuario en el terminal mediante el ratón o el teclado).

<sup>12</sup>Personal Use o Evaluation License

*dom0* y tiene ciertas características especiales respecto al resto, ya que es la única con privilegios para acceder de manera directa a todo el hardware, además de ser la única que puede realizar tareas de control sobre el propio Xen, como iniciar y parar el resto de máquinas virtuales.

**Ventajas:** buen rendimiento, abundante documentación.

**Desventajas:** instalación compleja, da problemas con algunos controladores, no es compatible con la ACPI (Interfaz avanzada de configuración y energía).

**Coste:** gratuito.

### 2.4.5. Qemu

QEMU es un emulador de CPU basado en la traducción dinámica de binarios (conversión del código binario de la arquitectura fuente en código entendible por la arquitectura huésped). QEMU también tiene capacidades de virtualización dentro de un sistema operativo, ya sea Linux, Windows, o cualquiera de los sistemas operativos admitidos. Esta máquina virtual puede ejecutarse en cualquier tipo de Microprocesador o arquitectura. QEMU posee dos modos de operación:

- Emulación del modo usuario: puede ejecutar procesos compilados para un tipo de procesador en otro tipo de procesador.
- Modo de emulación completo: emula un hardware completo, incluyendo CPU y varios periféricos.

La mayoría del programa está bajo licencia LGPL y el modo de emulación de usuario tiene licencia GPL.

**Ventajas:** código libre, fácil configuración.

**Desventajas:** soporte escaso, desarrollo irregular, velocidad de CPU muy baja en entornos emulados. El consumo de recursos es mejorable.

**Coste:** Coste: gratuito, es GPL.

## 2.5. Comparativas

Para ver realmente cuales son las máquinas virtuales más conocidas, utilizaremos una herramienta que nos proporciona Google<sup>13</sup>, *Google Trends*, para ver las más solicitadas. La idea es que normalmente recurrimos a buscadores como Google para informarnos de lo cosas que necesitamos o interesan, por lo que realizando estas comparativas de máquinas virtuales, deberíamos ver cuales de estas herramientas son las más conocidas, lo que nos permitiría extrapolar cuales de estas son las más utilizadas.

La figura 2.4 muestra las máquinas virtuales que podemos obtener de forma gratuita y tomamos como referencia a Vmware por su amplia difusión. A parte de las cinco que nos muestra (VMware, Xen, VirtualBox, OpenVZ y Use Mode Linux), también estaba incluida Linux-Server, pero tenía un volumen de búsqueda entorno al 0, por lo que ni tan siquiera salía en las gráficas. El eje horizontal de la gráfica representa el tiempo, y el eje vertical la frecuencia con la que se ha buscado el término globalmente. Se puede observar a primera vista como, en el volumen de búsquedas que se realiza en Google, Vmware es la que tiene más demanda y la solución de libre distribución que le sigue más de cerca es VirtualBox.

---

<sup>13</sup>Google Trends es una herramienta de Google Labs que muestra los términos de búsqueda más populares del pasado reciente. Las gráficas de Google Trends representan con cuanta frecuencia se realiza una búsqueda particular en varias regiones del mundo.

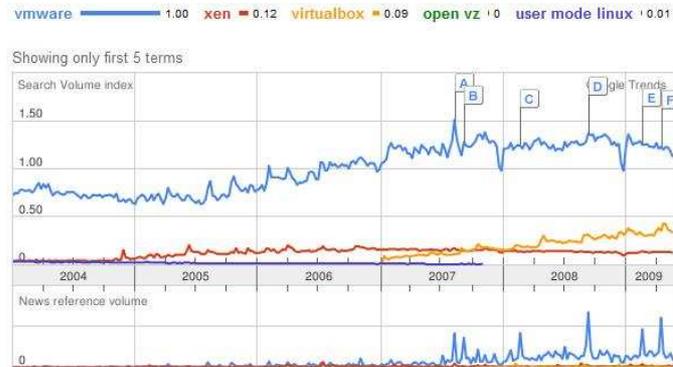


Figura 2.4: Resultado búsqueda VM's de software libre

En la figura 2.5 vemos los resultados de la búsqueda para las soluciones privadas. En esta gráfica también se puede observar el dominio de Vmware sobre otras soluciones como Microsoft Virtual PC y Parallels (máquina virtual muy utilizada en sistemas operativos de Mac).

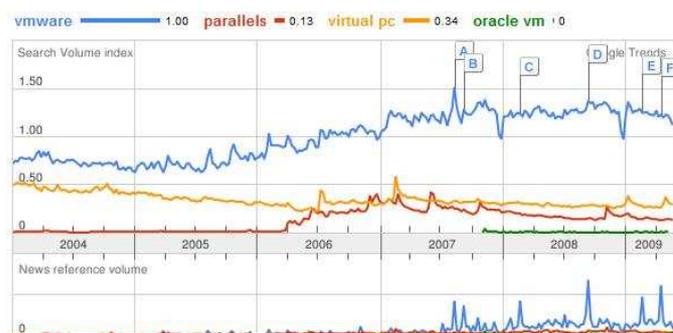


Figura 2.5: Resultado búsqueda VM's de software privado

## 2.6. Conclusiones

Vistos los resultados obtenidos en las figuras 2.4 y 2.5 y las ventajas y desventajas de las soluciones de virtualización anteriormente comentadas, hemos optado por realizar nuestros estudios de rendimiento con VMware y con la distribución de software libre VirtualBox. Con estas dos aplicaciones realizaremos las pruebas sobre procesador, memoria y red.

En el siguiente capítulo detallamos los resultados obtenidos con estas dos aplicaciones en las diferentes pruebas a las que las hemos sometido. Terminaremos por elegir una de ellas para que nos sirva de base para el desarrollo de nuestra aplicación que permite la creación, gestión y modificación de máquinas virtuales de una forma rápida y según las necesidades de los usuarios.

# Capítulo 3

## Estudio de rendimiento

En este capítulo vamos a realizar un estudio de rendimiento sobre las dos herramientas de virtualización elegidas en el capítulo anterior, VMware y Virtual-Box. Este estudio estará centrado en los siguientes elementos de nuestro computador: red, procesador y memoria. En las siguientes páginas vamos a mostrar los resultados obtenidos en cada una de las pruebas realizadas en nuestra máquina sobre cada uno de los puntos de estudio, estos son tráfico de red, memoria y procesador. Las características de la máquina que utilizaremos son las siguientes:

- Procesador: Intel Core 2 Quad Q6600 a 2.40 GHz.
- Memoria: 4 GB.
- Sistema Operativo: Windows Vista Business 32 bits.
- Conexión de red de la línea ADSL:
  - Velocidad de descarga : 5154,3 kbps (644,3 kB/seg, ratio de transferencia).
  - Velocidad de subida: 267 kbps (33,4 kB/seg, ratio de transferencia).

Con las siguientes pruebas, lo que se busca, es encontrar las posibles ventajas que nos pueda aportar el uso de máquinas virtuales. Empezaremos con las pruebas

de tráfico de datos por red utilizando un programa *peer to peer*<sup>1</sup>, seguiremos con los tests de procesador utilizando un programa de cálculo intensivo y para finalizar, las pruebas de memoria mediante un software de renderizado de imágenes.

## 3.1. Red

En este apartado se mostrarán los resultados obtenidos al someter tanto al *host* como a las diferentes máquinas virtuales, al proceso de descarga de archivos mediante software p2p. El objetivo es observar si con la utilización de máquinas virtuales se obtiene un rendimiento superior en lo que respecta al tráfico de red.

### 3.1.1. Primer juego de pruebas

Utilizaremos para estas pruebas un programa *peer to peer* llamado Azureus<sup>2</sup>. Las pruebas se repetirán 2 veces para nuestra máquina, para así poder hacer una media de los datos obtenidos y descartar posibles irregularidades durante alguno de los test. Al depender de un programa p2p los resultados de las pruebas pueden ser muy variables, ya que no solo depende de la máquina sino de las conexiones entrantes.

El experimento consiste en descargar los mismos archivos en nuestras máquinas. Se han elegido tres archivos pertenecientes a una distribución Debian, se dejará correr la aplicación durante 60 minutos y después se harán los cálculos de la velocidad media de descarga obtenida durante esa sesión.

---

<sup>1</sup>Una red *peer to peer* (o P2P), es una red de computadoras en la que todos o algunos aspectos de esta funcionan sin clientes ni servidores fijos, sino una serie de nodos que se comportan como iguales entre sí. Es decir, actúan simultáneamente como clientes y servidores respecto a los demás nodos de la red.

<sup>2</sup>Azureus es un cliente de BitTorrent de código abierto y desarrollado en lenguaje de programación Java, por lo que es multiplataforma (teniendo instalada la máquina virtual de Java) y que funciona tanto en sistemas Mac, como Windows o GNU/Linux.

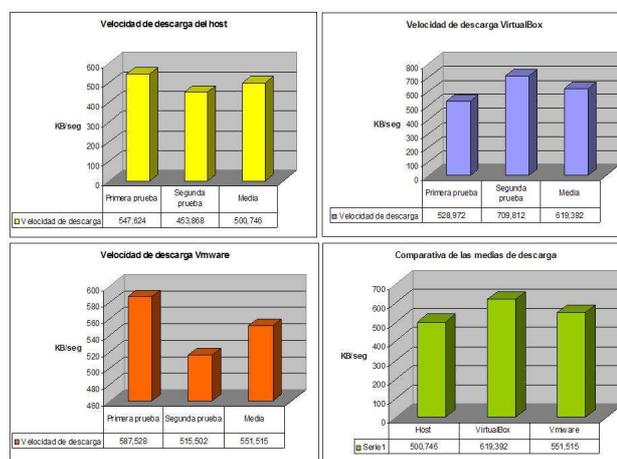


Figura 3.1: Resultado primer juego de pruebas p2p.

A continuación procedemos a la explicación del proceso de cada una de las pruebas, pero a modo de avance, en la figura 3.1 podemos observar los resultados ofrecidos en cada una de las máquinas virtuales y en el *host*, así como la comparativa de la media de las dos pruebas que se realizan por cada máquina. Se puede observar como, tanto, VMware como VirtualBox, sobrepasan la velocidad de descarga que nos ofrece nuestra conexión de red.

La explicación a que las VM tengan una mayor tasa de descarga que el anfitrión es la siguiente: al disponer en el mismo momento de dos máquinas virtuales descargando los mismos ficheros, estas comparten entre sí las partes de los archivos que la otra ya ha descargado, por lo que la velocidad de descarga se ve incrementada llegando a picos de 800 KB/seg. Esto sucede por encontrarse ambas máquinas dentro de la misma LAN<sup>3</sup>, por lo que la compartición de datos entre ellas produce que nos aumente la tasa de descarga, dando unos resultados esperados por

<sup>3</sup>Una red de área local o LAN (del inglés Local Area Network) es la interconexión de varios ordenadores y periféricos. Su extensión está limitada físicamente a un edificio o a un entorno de 200 metros o con repetidores podríamos llegar a la distancia de un campo de 1 kilómetro. Su aplicación más extendida es la interconexión de ordenadores personales y estaciones de trabajo en oficinas, fábricas, etc., para compartir recursos e intercambiar datos y aplicaciones. En definitiva, permite que dos o más máquinas se comuniquen.

la tipología de la red, pero no válidos para nuestro estudio.

A continuación, mostramos el proceso seguido en el transcurso del primer juego de pruebas para nuestro *host*, VirtualBox y VMware.

### Host

Durante la primera prueba obtenemos los datos mostrados en la tabla 3.1, donde Tiempo es el tiempo que tarda en descargarse el archivo:

	Tamaño del fichero	Descargado	Tiempo
<b>Fichero 1</b>	644.60 MB	644.50 MB	50 minutos
<b>Fichero 2</b>	637.43 MB	637.43 MB	54 minutos
<b>Fichero 3</b>	643.21 MB	643.21 MB	59 minutos

Tabla 3.1: Tasa de descarga obtenida en el host en la primera prueba.

$$TotalDescargado = 1925,24MB * \frac{1024kB}{1MB} = 1971445,76kB$$

$$Tiempo = 59min * \frac{60seg}{1min} = 3540seg$$

$$T_{md} = \frac{TotalDescargado}{Tiempoensegundos} = \frac{1971445,76kB}{3540seg} = 556,906kB/seg$$

; donde la Tasa media de descarga es  $T_{md}$ .

Si contamos una hora entera el resultado es de **547,624 kB/seg**.

En la segunda prueba obtenemos los datos mostrados en la tabla 3.2.

	Tamaño del fichero	Descargado	Tiempo
<b>Fichero 1</b>	644.60 MB	644.50 MB	53 minutos
<b>Fichero 2</b>	637.43 MB	637.43 MB	21 minutos
<b>Fichero 3</b>	643.21 MB	313.7 MB	60 minutos

Tabla 3.2: Tasa de descarga obtenida en el host en la segunda prueba.

$$TotalDescargado = 1595,63MB * \frac{1024kB}{1MB} = 1633925,12kB$$

$$T_{md} = \frac{TotalDescargado}{Tiempoensegundos} = \frac{1633925,12kB}{3600seg} = 453,868kB/seg.$$

Con las dos pruebas realizadas, hacemos la media de ambas obteniendo:

$$Media = \frac{Prueba1 + Prueba2}{2} = \frac{5470,624kB/seg + 453,868kB/seg}{2} = \mathbf{500,746\text{ kB/seg}}$$

### VirtualBox

Primera prueba: se lanzan 2 máquinas virtuales, con los mismos archivos que en las pruebas anteriores. En la tabla 3.3 vemos los resultado de la primera máquina virtual y en la tabla 3.4 los de la segunda VM.

	Tamaño del fichero	Descargado	Tiempo
<b>Fichero 1</b>	644.60 MB	233.33 MB	60 minutos
<b>Fichero 2</b>	637.43 MB	545.12 MB	60 minutos
<b>Fichero 3</b>	643.21 MB	206.96 MB	60 minutos

Tabla 3.3: Tasa de descarga obtenida la primera VM VirtualBox en la primera prueba.

#### Máquina Virtual 1

$$TotalDescargado = 985,41MB * \frac{1024kB}{1MB} = 1009059,84kB$$

$$T_{md} = \frac{1009059,84kB}{3600seg} = 280,294kB/seg$$

#### Máquina Virtual 2

$$TotalDescargado = 874,26MB * \frac{1024kB}{1MB} = 895242,24kB$$

$$T_{md} = \frac{895242,24kB}{3600seg} = 248,678kB/seg.$$

	<b>Tamaño del fichero</b>	<b>Descargado</b>	<b>Tiempo</b>
<b>Fichero 1</b>	644.60 MB	209.98 MB	60 minutos
<b>Fichero 2</b>	637.43 MB	523.94 MB	60 minutos
<b>Fichero 3</b>	643.21 MB	140.34 MB	60 minutos

Tabla 3.4: Tasa de descarga obtenida la segunda VM VirtualBox en la primera prueba.

Sumando la tasa de descarga en las dos máquinas virtuales obtenemos:

$$Media = 280,294kB/seg + 248,678kB/seg = 528,972kB/s$$

Obteniendo picos de descarga de 800 kB/seg en ambas máquinas virtuales.

Segunda prueba: en la tabla 3.5 vemos los resultados obtenidos en esta segunda prueba para la primera máquina. Y en la tabla 3.6 vemos los resultados obtenidos para la segunda máquina.

	<b>Tamaño del fichero</b>	<b>Descargado</b>	<b>Tiempo</b>
<b>Fichero 1</b>	644.60 MB	443.33 MB	60 minutos
<b>Fichero 2</b>	637.43 MB	637.43 MB	35 minutos
<b>Fichero 3</b>	643.21 MB	270.81 MB	60 minutos

Tabla 3.5: Tasa de descarga obtenida la primera VM VirtualBox en la segunda prueba.

#### **Máquina Virtual 1**

$$TotalDescargado = 1351,57MB * \frac{1024kB}{1MB} = 1384007,68kB$$

$$T_{md} = \frac{1384007,68kB}{3600seg} = 384,447kB/seg$$

#### **Máquina Virtual 2**

$$TotalDescargado = 1143,86MB * \frac{1024kB}{1MB} = 1171312,64kB$$

	<b>Tamaño del fichero</b>	<b>Descargado</b>	<b>Tiempo</b>
<b>Fichero 1</b>	644.60 MB	330.54 MB	60 minutos
<b>Fichero 2</b>	637.43 MB	637.43 MB	35 minutos
<b>Fichero 3</b>	643.21 MB	175.89 MB	60 minutos

Tabla 3.6: Tasa de descarga obtenida la segunda VM VirtualBox en la segunda prueba.

$$T_{md} = \frac{1171312,64kB}{3600seg} = 325,365kB/seg.$$

Sumando ambos datos obtenidos en esta prueba se obtiene:

$$Media = 384,447kB/seg + 325,365kB/seg = 709,812kB/seg$$

Una vez realizadas las 2 pruebas para VirtualBox calculamos la media:

$$Media = \frac{528,972kB/seg + 709,812kB/seg}{2} = 619,392kB/seg$$

## VMware

Primera prueba: se lanzan 2 máquinas virtuales, con los mismos archivos que en las pruebas anteriores. La tabla 3.7 vemos los resultados para la máquina 1, y en la tabla 3.8 pertenecientes ambas a VMware.

	<b>Tamaño del fichero</b>	<b>Descargado</b>	<b>Tiempo</b>
<b>Fichero 1</b>	644.60 MB	397.54 MB	60 minutos
<b>Fichero 2</b>	637.43 MB	468.12 MB	60 minutos
<b>Fichero 3</b>	643.21 MB	544.13 MB	60 minutos

Tabla 3.7: Tasa de descarga obtenida la primera VM VMware en la primera prueba.

### Máquina Virtual 1

$$TotalDescargado = 1409,79MB * \frac{1024kB}{1MB} = 1443624,96kB$$

$$T_{md} = \frac{1443624,96kB}{3600seg} = 401,007kB/seg.$$

	<b>Tamaño del fichero</b>	<b>Descargado</b>	<b>Tiempo</b>
<b>Fichero 1</b>	644.60 MB	345.35 MB	60 minutos
<b>Fichero 2</b>	637.43 MB	196.35 MB	60 minutos
<b>Fichero 3</b>	643.21 MB	114.05 MB	60 minutos

Tabla 3.8: Tasa de descarga obtenida la segunda VM VMware en la primera prueba.

### Máquina Virtual 2

$$TotalDescargado = 655,75MB * \frac{1024kB}{1MB} = 671488kB$$

$$T_{md} = \frac{671488kB}{3600seg} = 186,524kB/seg$$

La suma de descarga de las dos máquinas virtuales corriendo en paralelo:

$$Media = 401,007kB/seg + 186,524kB/seg = 587,528KB/s$$

Segunda prueba: se lanzan 2 máquinas virtuales, con los mismos archivos que en las pruebas anteriores. Los resultados de esta segunda prueba para VMware las podemos ver en las tablas 3.9 y 3.10.

	<b>Tamaño del fichero</b>	<b>Descargado</b>	<b>Tiempo</b>
<b>Fichero 1</b>	644.60 MB	644.50 MB	50 minutos
<b>Fichero 2</b>	637.43 MB	267.91 MB	60 minutos
<b>Fichero 3</b>	643.21 MB	428.20 MB	60 minutos

Tabla 3.9: Tasa de descarga obtenida la primera VM VMware en la segunda prueba.

### Máquina Virtual 1

$$TotalDescargado = 655,75MB * \frac{1024kB}{1MB} = 671488kB$$

$$T_{md} = \frac{1372887,04kB}{3600seg} = 381,358kB/seg$$

	Tamaño del fichero	Descargado	Tiempo
<b>Fichero 1</b>	644.60 MB	219.42 MB	60 minutos
<b>Fichero 2</b>	637.43 MB	123.25 MB	60 minutos
<b>Fichero 3</b>	643.21 MB	128.93 MB	60 minutos

Tabla 3.10: Tasa de descarga obtenida la segunda VM VMware en la segunda prueba.

### Máquina Virtual 2

$$TotalDescargado = 471,6MB * \frac{1024kB}{1MB} = 482918,4kB$$

$$T_{md} = 1 \frac{482918,4kB}{3600seg} = 134,144kB/seg$$

La suma de descarga de las dos máquinas virtuales corriendo en paralelo:

$$Media = 381,358kB/seg + 134,144kB/seg = 515,502kB/seg$$

La media de las dos pruebas es:

$$Media = \frac{587,528kB/seg + 515,502kB/seg}{2} = 551,515kB/seg$$

En la tabla 3.11 podemos ver de forma resumida los resultados obtenidos en las pruebas realizadas anteriormente. Como se puede comprobar, la media de descarga de las herramientas de virtualización VMware y VirtualBox, superan la media de nuestra máquina anfitriona.

	Primera Prueba	Segunda Prueba	Media
<b>Host</b>	556.906 KB/seg	453.868 KB/seg	505.387 KB/seg
<b>VMware</b>	587.528 KB/seg	515.502 KB/seg	551.515 KB/seg
<b>VirtualBox</b>	528.972 KB/seg	709.812 KB/seg	619.392 KB/seg

Tabla 3.11: Resultados del primer juego de pruebas.

En la figura 3.2 vemos gráficamente como las máquinas virtuales han obtenido una mayor tasa de descarga que el *host*.

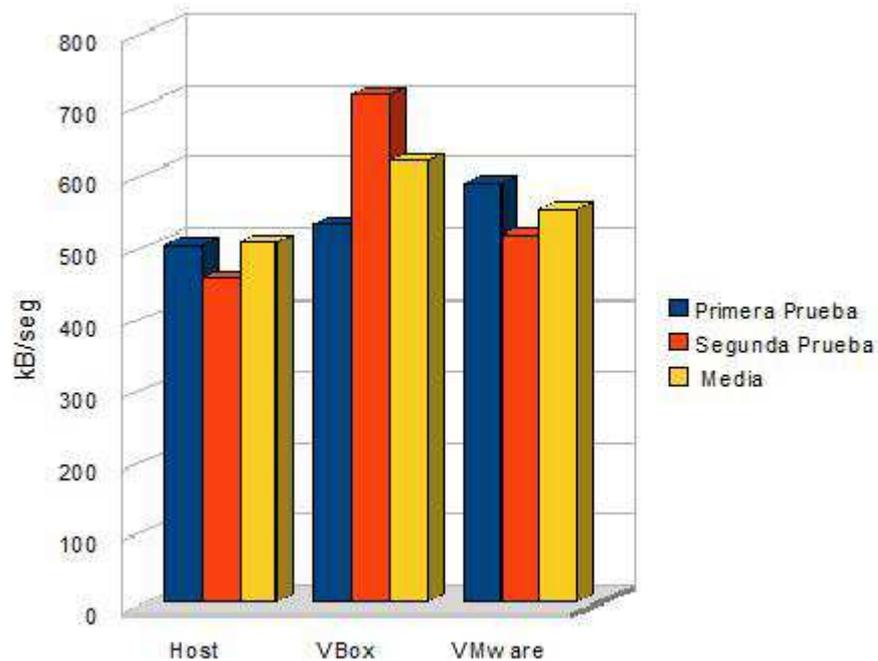


Figura 3.2: Gráfica de los resultados anteriores.

## Conclusiones

Como se ha comentado anteriormente y ahora hemos corroborado, el hecho de tener ambas máquinas dentro del mismo equipo y de la misma LAN, ha provocado un aumento en la tasa de descarga global al compartir entre ellas los archivos ya descargados, alcanzando picos de descarga de 1.4 MB/seg. Por tanto, una vez realizado el primer juego de pruebas, realizaremos un segundo juego de pruebas con archivos diferentes para eliminar esto.

### 3.1.2. Segundo juego de pruebas

Visto que en las pruebas anteriores, al compartir las dos máquinas virtuales los archivos a descargar, compartían entre ellas las partes ya descargadas, ahora vamos a poner archivos diferentes en las dos máquinas virtuales. Con esto, pretendemos ver realmente la tasa de descarga que podemos obtener sin esa compartición interna que hemos observado anteriormente.

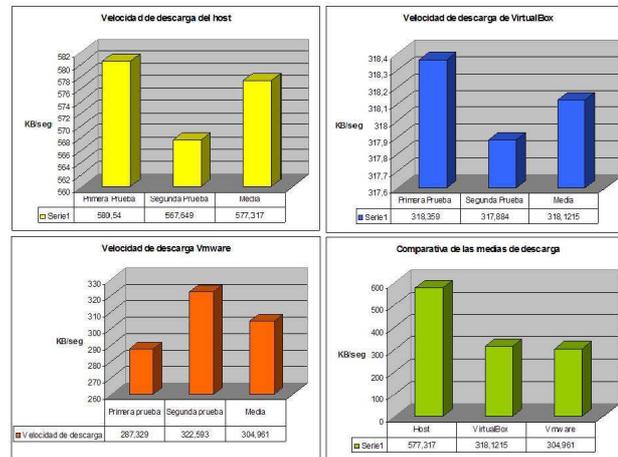


Figura 3.3: Resumen del segundo juego de prueba p2p.

En la figura 3.3 podemos observar como la tasa de descarga baja drásticamente comparándola con la del primer juego de pruebas en la que existía una compartición entre las máquinas virtuales.

En las siguientes líneas se explica el proceso que se ha llevado acabo para la realización de este segundo juego de pruebas. Empezaremos con el *host*, para después continuar con VMware y VirtualBox.

## Host

En la tabla 3.12 podemos ver los resultados obtenidos en la primera prueba realizada por el anfitrión con los seis archivos elegidos. Y en la tabla 3.13 encontramos los resultados del segundo test realizado sobre el *host*.

Primera prueba:

$$TotalDescargado = 2040,96MB * \frac{1024kB}{1MB} = 2089943,04kB$$

$$T_{md} = \frac{2089943,04kB}{3600seg} = 580,540kB/seg$$

	<b>Tamaño del fichero</b>	<b>Descargado</b>	<b>Tiempo</b>
<b>Fichero 1</b>	696.81 MB	498.20 MB	60 minutos
<b>Fichero 2</b>	698.82 MB	25.50 MB	60 minutos
<b>Fichero 3</b>	637.31 MB	22.44 MB	60 minutos
<b>Fichero 1</b>	644.60 MB	435.78 MB	60 minutos
<b>Fichero 2</b>	637.43 MB	571.27 MB	60 minutos
<b>Fichero 3</b>	643.21 MB	487.77 MB	60 minutos

Tabla 3.12: Tasa de descarga obtenida por el host en la primera prueba.

Segunda prueba:

	<b>Tamaño del fichero</b>	<b>Descargado</b>	<b>Tiempo</b>
<b>Fichero 1</b>	696.81 MB	376.68 MB	60 minutos
<b>Fichero 2</b>	698.82 MB	132.10 MB	60 minutos
<b>Fichero 3</b>	637.31 MB	172.34 MB	60 minutos
<b>Fichero 1</b>	644.60 MB	352.59 MB	60 minutos
<b>Fichero 2</b>	637.43 MB	502.32 MB	60 minutos
<b>Fichero 3</b>	643.21 MB	459.61 MB	60 minutos

Tabla 3.13: Tasa de descarga obtenida por el host en la segunda prueba.

$$TotalDescargado = 1995,64MB * \frac{1024kB}{1MB} = 2043535,36kB$$

$$T_{md} = \frac{2043535,36kB}{3600seg} = 567,649kB/seg$$

La media de las dos pruebas es:

$$Media = \frac{580,540kB/seg + 567,649kB/seg}{2} = 574,0945kB/seg$$

## VMware

Primera prueba: en la tabla 3.14 vemos los resultados de la primera máquina de VMware. Mientras que en la tabla 3.15 observamos los resultados de la segunda máquina virtual. Como se puede ver, los datos descargados han caído drásticamente comparándolos con los del primer juego de pruebas.

	<b>Tamaño del fichero</b>	<b>Descargado</b>	<b>Tiempo</b>
<b>Fichero 1</b>	644.60 MB	64.95 MB	60 minutos
<b>Fichero 2</b>	637.43 MB	479.70 MB	60 minutos
<b>Fichero 3</b>	643.21 MB	214.78 MB	60 minutos

Tabla 3.14: Tasa de descarga obtenida por la primera VM VMware en la primera prueba.

### Máquina Virtual 1

$$TotalDescargado = 759,43MB * \frac{1024kB}{1MB} = 777656,32kB$$

$$T_{md} = \frac{777656,32kB}{3600seg} = 216,016kB/seg$$

	<b>Tamaño del fichero</b>	<b>Descargado</b>	<b>Tiempo</b>
<b>Fichero 1</b>	696.81 MB	86.76 MB	60 minutos
<b>Fichero 2</b>	698.82 MB	71.43 MB	60 minutos
<b>Fichero 3</b>	637.31 MB	92.52 MB	60 minutos

Tabla 3.15: Tasa de descarga obtenida por la segunda VM VMware en la primera prueba.

### Máquina Virtual 2

$$TotalDescargado = 250,71MB * \frac{1024kB}{1MB} = 256727,04kB$$

$$T_{md} = \frac{256727,04kB}{3600seg} = 71,313kB/seg$$

La suma de descarga de las dos máquinas virtuales corriendo en paralelo:

$$Media = 216,016KB/seg + 71,313KB/seg = 287,329KB/seg$$

Segunda prueba: en las tablas 3.16 y 3.17 vemos los resultados de la segunda tanda de pruebas para VMware.

	<b>Tamaño del fichero</b>	<b>Descargado</b>	<b>Tiempo</b>
<b>Fichero 1</b>	644.60 MB	338.36 MB	60 minutos
<b>Fichero 2</b>	637.43 MB	637.43 MB	43 minutos
<b>Fichero 3</b>	643.21 MB	39.80 MB	60 minutos

Tabla 3.16: Tasa de descarga obtenida por la primera VM VMware en la segunda prueba.

### Máquina Virtual 1

$$TotalDescargado = 1015,59MB * \frac{1024kB}{1MB} = 1039964,16kB$$

$$T_{md} = \frac{1039964,16kB}{3600seg} = 288,878kB/seg$$

	<b>Tamaño del fichero</b>	<b>Descargado</b>	<b>Tiempo</b>
<b>Fichero 1</b>	696.81 MB	40.30 MB	60 minutos
<b>Fichero 2</b>	698.82 MB	35.12 MB	60 minutos
<b>Fichero 3</b>	637.31 MB	43.11 MB	60 minutos

Tabla 3.17: Tasa de descarga obtenida por la segunda VM VMware en la segunda prueba.

### Máquina Virtual 2

$$TotalDescargado = 118,53MB * \frac{1024kB}{1MB} = 121374,72kB$$

$$T_{md} = \frac{121374,72kB}{3600seg} = 33,715kB/seg$$

La suma de descarga de las dos máquinas virtuales corriendo en paralelo:

$$Media = 288,878kB/seg + 33,715kB/seg = 322,593kB/seg$$

La media de las dos pruebas es:

$$Media = \frac{287,329kB/seg + 322,593kB/seg}{2} = 304,961kB/seg$$

### VirtualBox

Primera prueba: en las tablas 3.18 y 3.19 vemos los resultados obtenidos en las respectivas máquinas pertenecientes a VirtualBox.

	Tamaño del fichero	Descargado	Tiempo
<b>Fichero 1</b>	644.60 MB	322.94 MB	60 minutos
<b>Fichero 2</b>	637.43 MB	431.97 MB	60 minutos
<b>Fichero 3</b>	643.21 MB	176.71 MB	60 minutos

Tabla 3.18: Tasa de descarga obtenida por la primera VM VirtualBox en la primera prueba.

#### Máquina Virtual 1

$$TotalDescargado = 931,62MB * \frac{1024kB}{1MB} = 953978,88kB$$

$$T_{md} = \frac{953978,88kB}{3600seg} = 264,994kB/seg$$

	Tamaño del fichero	Descargado	Tiempo
<b>Fichero 1</b>	696.81 MB	64.91 MB	60 minutos
<b>Fichero 2</b>	698.82 MB	56.49 MB	60 minutos
<b>Fichero 3</b>	637.31 MB	66.21 MB	60 minutos

Tabla 3.19: Tasa de descarga obtenida por la segunda VM VirtualBox en la primera prueba.

#### Máquina Virtual 2

$$TotalDescargado = 187,61MB * \frac{1024kB}{1MB} = 192112,64kB$$

$$T_{md} = \frac{192112,64kB}{3600seg} = 53,365kB/seg$$

La suma de descarga de las dos máquinas virtuales corriendo en paralelo:

$$Media = 264,994kB/seg + 53,365kB/seg = 318,359kB/seg$$

Segunda prueba: en las tablas 3.20 y 3.21 vemos los resultados obtenidos en las respectivas máquinas pertenecientes a VirtualBox.

	<b>Tamaño del fichero</b>	<b>Descargado</b>	<b>Tiempo</b>
<b>Fichero 1</b>	644.60 MB	371.15 MB	60 minutos
<b>Fichero 2</b>	637.43 MB	442.53 MB	60 minutos
<b>Fichero 3</b>	643.21 MB	123.10 MB	60 minutos

Tabla 3.20: Tasa de descarga obtenida por la primera VM VirtualBox en la primera prueba.

### Máquina Virtual 1

$$TotalDescargado = 936,78MB * \frac{1024kB}{1MB} = 959262,72kB$$

$$T_{md} = \frac{959262,72kB}{3600seg} = 266,462kB/seg$$

	<b>Tamaño del fichero</b>	<b>Descargado</b>	<b>Tiempo</b>
<b>Fichero 1</b>	696.81 MB	69.51 MB	60 minutos
<b>Fichero 2</b>	698.82 MB	43.89 MB	60 minutos
<b>Fichero 3</b>	637.31 MB	67.38 MB	60 minutos

Tabla 3.21: Tasa de descarga obtenida por la segunda VM VirtualBox en la primera prueba.

### Máquina Virtual 2

$$TotalDescargado = 180,78MB * \frac{1024kB}{1MB} = 185118,72kB$$

$$T_{md} = \frac{185118,72kB}{3600seg} = 51,422kB/seg$$

La suma de descarga de las dos máquinas virtuales corriendo en paralelo:

$$Media = 266,462kB/seg + 51,422kB/seg = 317,884kB/seg$$

	Primera Prueba	Segunda Prueba	Media
<b>Host</b>	580.540 KB/seg	574.0945 KB/seg	577.317 KB/seg
<b>VMware</b>	287.329 KB/seg	322.593 KB/seg	304.961 KB/seg
<b>VirtualBox</b>	318.359 KB/seg	317.884	318.1215 KB/seg

Tabla 3.22: Resultados del segundo juego de pruebas.

Como se puede observar en la tabla 3.22, después de realizar este segundo juego de pruebas, no se obtiene ninguna ventaja directa utilizando máquinas virtuales corriendo en paralelo descargando archivos mediante un programa *peer to peer*, respecto al hacerlo directamente sobre el anfitrión (*host*). Esto es debido a que las máquinas virtuales comparten el canal de salida, que en la máquina en las que se realizaron las pruebas es único, por lo que tienen que competir por este recurso, con lo que en vez de obtener una mejora, obtenemos claramente una pérdida de rendimiento por lo que respecta al tráfico de red.

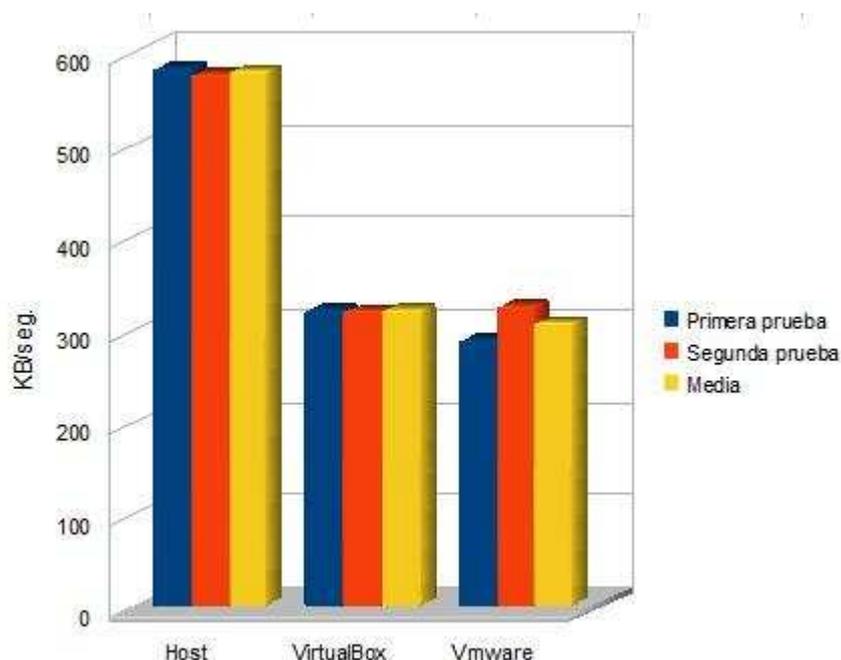


Figura 3.4: Gráfica de los resultados del segundo juego de pruebas

## 3.2. CPU

Para realizar las pruebas de rendimiento de procesador vamos a utilizar la herramienta `msieve`, como ya habíamos mencionado con anterioridad, para factorizar un número en sus dos primeros. Para crear un número  $n$ , con dos números primos de igual fortaleza utilizaremos la siguiente instrucción:

```
>openssl genrsa -out key.pem X
```

Donde  $X$  son los números de bits. La instrucción para mostrarlo es:

```
>openssl rsa -in key.pem -text -noout
```

En modulus tenemos la  $n$  y  $prime\ 1$ ,  $prime\ 2$  que son los primeros que nos devolverá al factorizar  $n$ .

Los números elegidos para las pruebas han sido elegidos con diferente longitud de bits para mostrar como un pequeño incremento en la longitud del número, aumenta mucho el tiempo de cálculo de sus dos primeros (con la carga sobre el procesador que ello supone). Estos números son los siguientes:

- Número A:
  - Número de dígitos: 73
  - Valor hexadecimal: 0x00a9c64c83e9920a5b7eafb5b1f31d42deef42dfe8ba934d061b69034dd36b
  
- Número B:
  - Número de dígitos: 82
  - Valor hexadecimal: 0x00f6529a6ce881e7e7b9742efe627c5a41ce1885244e9a2839ee524eb8f4ceeabc8219
  
- Número C:
  - Número de dígitos: 86
  - Valor hexadecimal: 0x11248298503889518007768935777016000978153419613999817286329668952672757390241894682937

- Número D:
  - Número de dígitos: 90
  - Valor hexadecimal: 0x244483849650351268321059272393156316297429768301683259376366343273352137041022159550036993

### Dificultad y complejidad de la factorización

Si un número grande, de  $b$  bits es el producto de dos primos de aproximadamente el mismo tamaño, no existe algoritmo conocido capaz de factorizarlo en tiempo polinomial. Esto significa que ningún algoritmo conocido puede factorizarlo en tiempo  $O(b^k)$ , para cualquier constante  $k$ . Aunque, existen algoritmos que son más rápidos que  $O(a^b)$  para cualquier  $a$  mayor que 1. En otras palabras, los mejores algoritmos son super-polinomiales, pero sub-exponenciales. De aquí el alto coste de uso que realizará el procesador al necesitar gran parte de su capacidad para realizar los cálculos necesarios para encontrar esos números primos que conforman  $b$ .

#### 3.2.1. Resultado de las pruebas de factorización

Para la realización de este test se repiten tres veces las pruebas para cada número y máquina para, posteriormente, hacer la media de los tiempos que han tardado en realizarse cada uno de los cálculos. En la tabla 3.23 se muestran los tiempos conseguidos en nuestra máquina.

	<b>A</b>	<b>B</b>	<b>C</b>	<b>D</b>
<b>Host</b>	129 seg.	1132.33 seg.	1653 seg.	3497 seg.
<b>VMware</b>	159 seg.	1384.33 seg.	1975 seg.	4406.5seg.
<b>VirtualBox</b>	126.67 seg.	1439 seg.	1992.33 seg.	4571.5 seg.

Tabla 3.23: Tiempo promedio de factorización.

Utilizando la virtualización, los tiempos de cálculo en la factorización de los diferentes números elegidos son mayores que los tiempos que tardó la máquina

*host*, por lo que podemos decir que la virtualización tampoco mejora el rendimiento del procesador en trabajos de cálculo intensivo, como se puede observar más claramente en la figura 3.5.

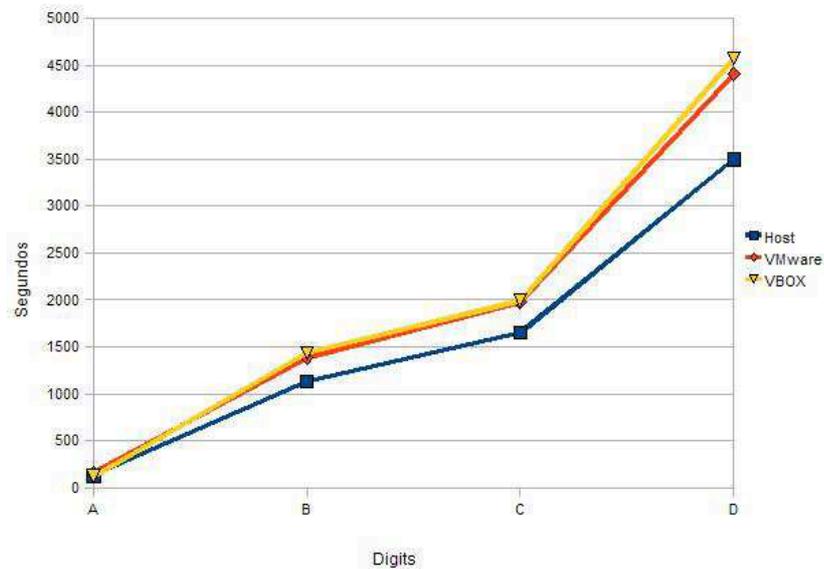


Figura 3.5: Comparativa de los tiempos obtenidos

### 3.3. Memoria

Para el estudio del rendimiento de la memoria en las máquinas virtuales, vamos a utilizar la aplicación Blender<sup>4</sup> para renderizar imágenes. El juego de pruebas consistirá en medir los tiempos que tarda la máquina anfitriona en realizar un vídeo de 250 frames, 750 frames y 1500 frames, y posteriormente ver si las

<sup>4</sup>Blender es un programa multiplataforma, dedicado especialmente al modelado, animación y creación de gráficos tridimensionales. El programa fue inicialmente distribuido de forma gratuita pero sin el código fuente, con un manual disponible para la venta, aunque posteriormente pasó a ser software libre.

máquinas virtuales de alguna manera nos rebajan esos tiempos consiguiendo una mejora general del rendimiento en este campo de estudio.

En la tabla 3.24 vemos el tiempo que ha tardado la máquina anfitriona en realizar el renderizado de los frames seleccionados en cada caso. Estos tiempos nos servirán de comparativa con los que obtengamos con las máquinas virtuales.

Host	250 frames	750 frames	1500 frames
	2 min 33 seg	8 min 20 seg	15 min 30 seg

Tabla 3.24: Tiempos de renderizado en la máquina host.

Ahora realizaremos la misma prueba para VirtualBox. Como se puede apreciar en la tabla 3.25, los tiempos se disparan respecto a los obtenidos por el anfitrión.

VirtualBox	250 frames	750 frames	1500 frames
1 VM	12 min 19 seg	41 min 25 seg	–

Tabla 3.25: Tiempos de renderizado con VirtualBox.

Con VMware, si miramos la tabla 3.26, podemos observar como utilizando una única máquina no obtenemos una mejora del tiempo de cómputo, sino que tarda más que el anfitrión (pero menos que VirtualBox). Por lo que seguiremos testeando únicamente con VMware.

Decidimos dividir la carga de trabajo en partes iguales, procesando cada parte en una VM, para repartir la carga del proceso de renderizado. Este reparto de carga se hace dividiendo en partes iguales el número de frames que va a procesar cada máquina virtual, para intentar que ambas VM's acaben aproximadamente al mismo tiempo y así intentar que ambas utilicen por igual el procesador.

El proceso de división será el siguiente: primero los 1500 frames los repartimos en 2 VM's con 750 frames cada una y se puede ver como los tiempos obtenidos son más parecidos a los que se pueden ver en el resultado de 1 máquina calculando 750 frames, que no con el esperado de 1 máquina calculando 1500 frames. Visto esta mejora, seguimos dividiendo la carga en 3 máquinas virtuales, y vemos una disminución del tiempo de renderizado, obteniendo tiempos parecidos a la de la máquina anfitriona.

VMware	250 frames	750 frames	1500 frames
1 VM	7 min 10 seg	23 min 49 seg	–

Tabla 3.26: Tiempos de renderizado con VMware.

VMware	1500 frames				
1 VM	---				
2 VM	27' 25"		21' 53"		
3 VM	16' 00"	14' 50"	17' 31"		
4 VM	12' 7"	15' 5"	9' 5"	12' 37"	
5 VM	11' 20"	12' 0"	9' 55"	10' 50"	11' 45"

Figura 3.6: Tiempos de renderizado de VMware utilizando varias VM's.

Dividiendo los 1500 frames en 4 VM's ya hemos conseguido rebajar el tiempo que tardaba el host en realizar este cálculo, pero el rendimiento de la máquina aún nos permite añadir una quinta máquina virtual. Con 5 máquinas trabajando en paralelo sobre los 1500 frames originales, cada una con su parte correspondiente, vemos como podemos renderizar los 1500 frames en un tiempo significativamente inferior que en el host utilizando VMware (figura 3.6).

Este último tiempo que parecería una posible mejora, no lo es si contamos, que en el *host*, aún habiendo realizado el renderizado en un tiempo algo superior que con cinco máquinas virtuales, tenemos el vídeo completo. No siendo el caso de las VM's, a ese tiempo deberíamos sumarle el tiempo que se tardaría en unir todas las partes que conforman el trabajo final.

### 3.4. Conclusiones

En los tres apartados anteriores hemos visto como la virtualización, a día de hoy, no nos permite mejorar el rendimiento de una única máquina, repartiéndola en diferentes máquinas virtuales. Después de realizar las pruebas anteriores y observar que las dos tecnologías de virtualización elegidas no ofrecían una mejora

sustancial en ninguna de los tres puntos de estudio anteriores (procesador, memoria y tráfico de red), se llega a la conclusión que, en el estado en que se encuentra la virtualización, la ventaja más importante que nos ofrece esta tecnología se encuentra en la utilización concurrente de usuarios sobre un anfitrión. Otra posible ventaja que nos puede representar el uso de diferentes máquinas virtuales es el ahorro energético (se estima un ahorro del 30 % [18]), que nos permite diferentes máquinas sobre la misma computadora.

Visto esto, hemos decidido realizar una aplicación que nos gestione la creación y modificación de máquinas virtuales (con diferente sistema operativo y de diferente tipo) sobre un *host*, de forma que, los usuarios puedan tener de una forma fácil y sin mayores costes económicos de, por ejemplo, servidores Web totalmente independientes entre si en una única máquina física. Esto es, juntar en una única máquina, servicios que de otra forma deberían estar ejecutándose en máquinas diferentes.



# Capítulo 4

## Diseño y desarrollo de la aplicación

Una vez realizados los estudios de rendimiento sobre los tres componentes fundamentales del ordenador (memoria, procesador y red) en los que no hemos obtenido un motivo cuantitativo para su utilización, en este capítulo vamos a pasar al diseño y desarrollo de nuestra aplicación. Se trata de un servidor personalizado de máquinas virtuales, para diferentes usuarios y sistemas operativos.

### 4.1. Especificación de requisitos

Con esta aplicación buscamos crear una herramienta para la creación y gestión de máquinas virtuales mediante una aplicación Web, de forma totalmente transparente y guiado para el usuario. A continuación veremos el software que ha sido utilizado durante el desarrollo de la aplicación, así como los motivos de la elección de este software.

#### 4.1.1. Software de virtualización

Para realizar esta aplicación hemos escogido VirtualBox (versión 1.6.6), pese a que VMware ha sido ligeramente superior a VirtualBox en las pruebas realizadas, para gestionar la virtualización de nuestra aplicación. Los motivos de esta elección son los siguientes:

- Nos proporciona un repertorio de comandos, lanzados en un terminal, muy amplio con el que poder crear/clonar, modificar, listar, etc las máquinas virtuales.
- Ofrece ejecución en forma remota de las máquinas mediante el protocolo RDP (Remote Desktop Protocol).
- Tiene un rango de configuración de las máquinas amplio.
- Al ser software libre y en consecuencia de código abierto, nos permite tener acceso a las fuentes y no nos implica ningún coste su utilización.

Para utilizar el escritorio remoto, VirtualBox nos ofrece tres posibilidades en la configuración de conexión [19], por lo que respecta al método de autenticación. Estas opciones que nos ofrece son las siguientes:

1. Null: este método no provee ningún tipo de autenticación, cualquier cliente podrá conectarse al servidor VRDP y así a la máquina virtual. Este es un método inseguro y sólo recomendado para redes privadas.
2. Externo: este método provee autenticación externa a través de librerías especiales de autenticación.

VirtualBox viene con dos librerías por defecto para autenticación externa:

- Para anfitriones Linux, *VRDPAuth.so*, autentifica los usuarios contra el sistema PAM<sup>1</sup> del anfitrión.
- En anfitriones Windows, *VRDPAuth.dll*, autentifica los usuarios contra el sistema WinLogon<sup>2</sup>.

En otra palabras, el método externo, por defecto lleva a cabo la autenticación con las cuentas de usuarios existentes en el sistema anfitrión. Sin embargo, es posible reemplazar este módulo de autenticación con cualquier

---

<sup>1</sup>PAM (Pluggable Authentication Modules) es una suite de librerías compartidas que permiten al administrador del sistema escoger cómo autentican a los usuarios las aplicaciones.

<sup>2</sup>WinLogon es un proceso que se encarga de validar la identidad de un usuario en el sistema.

otro módulo. Para esto, VirtualBox provee una interfaz que permite escribir tú propio módulo de autenticación.

3. Huésped: este método realiza la autenticación con un componente especial que viene con el *Guest Additions*; por lo que la autenticación no es llevada a cabo con los usuarios huéspedes, sino con las cuentas del usuario anfitrión. Este método, actualmente, sigue en fase de pruebas y todavía no está soportado.

Una vez comentados los tres tipos de autenticación que nos proporciona VirtualBox y descartando de entrada el método huésped por estar aún en fase de pruebas, nos quedan los métodos null y externo. En la realización de esta aplicación se ha optado por el uso del método null, por un motivo, y es que el método externo requiere que los usuarios que quieran utilizar la interfaz remota dispongan también de una cuenta de usuario en la máquina anfitriona. El método externo sería ideal para entornos con un grupo de usuarios predefinido, pero por el momento no para esta aplicación, en la que no se han definido un grupo concreto de usuarios a los que va destinado el uso de la aplicación.

#### 4.1.2. Software adicional

Para la realización de la aplicación Web utilizaremos, a parte de VirtualBox, un servidor HTTP Apache. Se ha elegido este servidor Web por ser de libre distribución, multiplataforma (hay versiones para UNIX, distribuciones Linux, Windows y MAC), extensible (acepta módulos externos como Perl, PHP, Python, etc ) y por ser el servidor Web más utilizado actualmente.

Respecto al lenguaje de programación utilizados, hemos trabajado con PHP, HTML y Javascript. Se ha elegido utilizar PHP porque será el servidor, el encargado de interpretar gran parte de los procesos necesarios para el correcto funcionamiento de nuestra aplicación.

### 4.1.3. Hardware

Como hardware, hemos tenido disponible una máquina con las siguientes características:

- Procesador: Intel(R) Core(TM)2 Quad Q9300 a 2.50 GHz
- Memoria RAM: 8 GB
- Conexión de red:
  - Download speed: 7739 kbps (967.4 KB/sec transfer rate).
  - Upload Speed: 3877 kbps (484.6 KB/sec transfer rate).

Nuestra máquina tiene como sistema operativo una distribución Debian Lenny Testing. El kernel utilizado es el 2.6.26-1-686-bigmem SMP<sup>3</sup>.

## 4.2. Diseño de la aplicación

Empezaremos con una visión general de la aplicación. En la imagen 4.1 se muestra de forma resumida cada una de las partes en que estará compuesta la aplicación: crearVM, encender/apagar VM, modificar y estado. A continuación describimos las funcionalidades y características de cada una de ellas.

---

<sup>3</sup>La arquitectura SMP (Multi-procesamiento simétrico), se caracteriza por el hecho de que varios microprocesadores comparten el acceso a la memoria. Todos los microprocesadores compiten en igualdad de condiciones por dicho acceso, de ahí la denominación simétrico.

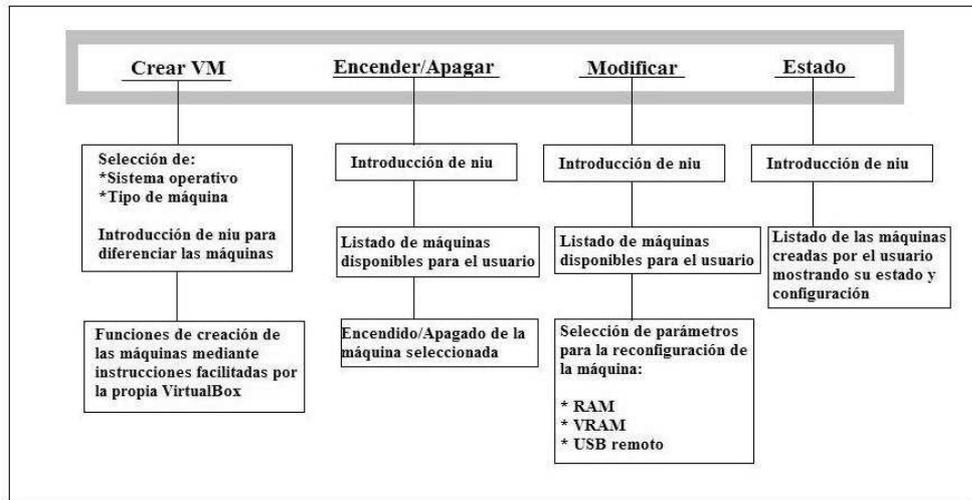


Figura 4.1: Resumen aplicación.

### 4.2.1. Creación de la máquina virtual

La figura 4.2 muestra el diagrama de la sección dedicada a la creación de las máquinas virtuales. Antes de iniciar el proceso de creación de una nueva VM, se comprueba el espacio libre en el disco y la memoria disponibles (esto corresponde a los dos primeros pasos, comprobación de disco y RAM). Si se ha llegado a un punto de un alto porcentaje de uso de uno de estos dos elementos, no se permitirá la clonación de más máquinas virtuales. Se ha elegido hacer estas comprobaciones para asegurar el correcto funcionamiento del proceso de clonación. Una vez que el sistema realiza estas comprobaciones, se permite al usuario elegir el tipo de máquina virtual más acorde con sus necesidades de entre todas las disponibles.

Cuando el usuario decide el tipo de VM que necesita, se lanza la orden de clonación de esa imagen. El proceso de clonación puede durar varios minutos (4 minutos aproximadamente), dependiendo del tamaño de disco a clonar, y se ha diseñado la aplicación para ir informando al usuario del estado de este proceso. Una vez clonada la imagen de disco se procede a la creación del archivo de configuración de la nueva máquina, otorgándole los parámetros necesarios para cada

tipo de VM.

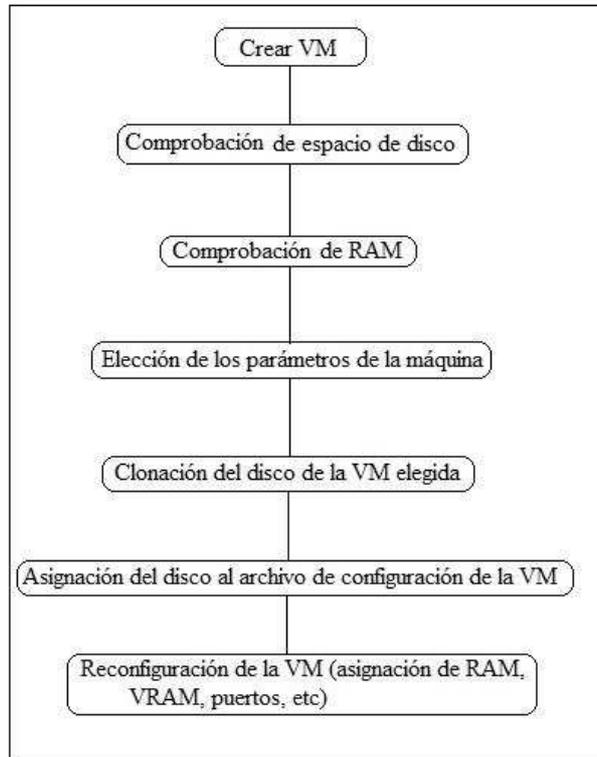


Figura 4.2: Diagrama de flujo de la creación de VM.

### 4.2.2. Control de encendido y apagado de la máquina virtual

En un segundo punto tenemos el encendido y apagado de las máquinas. Para poder utilizarlas remotamente, será necesario encenderlas mediante la aplicación Web, no así el apagado que se puede realizar directamente desde la Máquina virtual, pero se ha elegido dar la opción de apagar la VM desde la aplicación. En la figura 4.3 podemos ver el diagrama de esta parte de la aplicación.

A partir de identificador de usuario introducido, se listan las máquinas virtuales que posee el usuario. Una vez tenemos las máquinas del usuario, se procede a acceder al archivo de configuración de cada una de ellas para ver en que estado se encuentran las máquinas. Según el estado en que se encuentren, se mostrará la opción de iniciar la máquina (la máquina virtual estaba previamente apagada) o

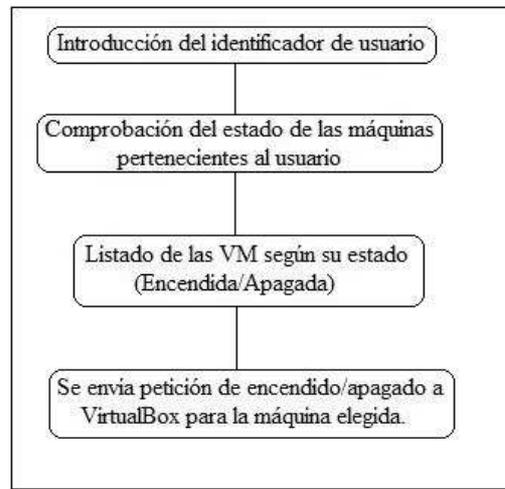


Figura 4.3: Diagrama de flujo de Encendido/Apagado de las VM.

en su defecto la opción de detenerla.

### 4.2.3. Modificación de las máquinas virtuales

El diagrama de la modificación de las diferentes máquinas virtuales lo podemos observar en la figura 4.4. Hay algunos parámetros que no podremos reconfigurar, como son el tamaño del disco, o el puerto asignado (esto es para evitar posibles conflictos con otras máquinas. Pero si se nos permitirá cambiar parámetros como la cantidad de memoria RAM o VRAM (memoria de vídeo) que utilizará nuestra máquina.

### 4.2.4. Estado de las máquinas virtuales

Por último, en la figura 4.5, podemos ver el diagrama de la sección de estado. Este es un apartado informativo, se busca que el usuario pueda tener siempre disponible información del estado en que se encuentran sus máquinas(encendida/apagada) o el puerto mediante el cual puede hacer la conexión remota.

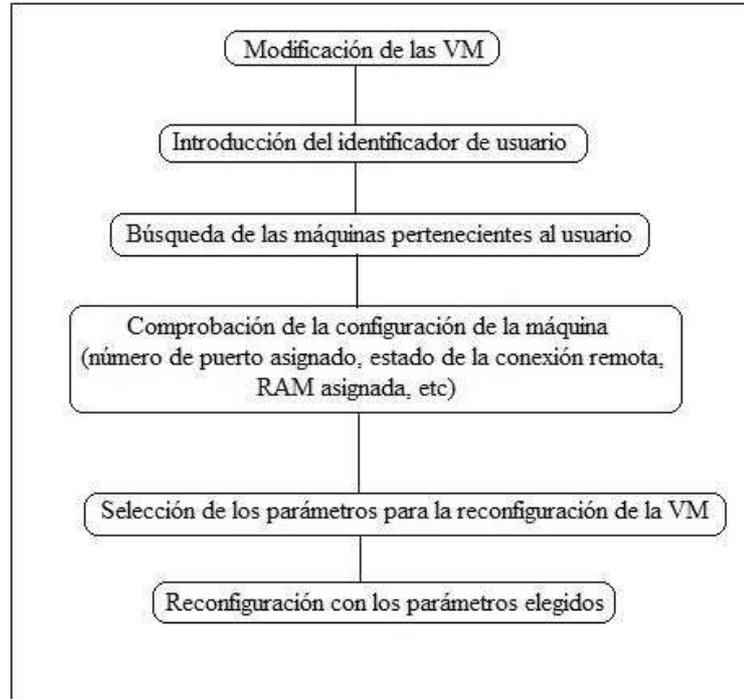


Figura 4.4: Diagrama de la sección de modificación de las máquinas virtuales.

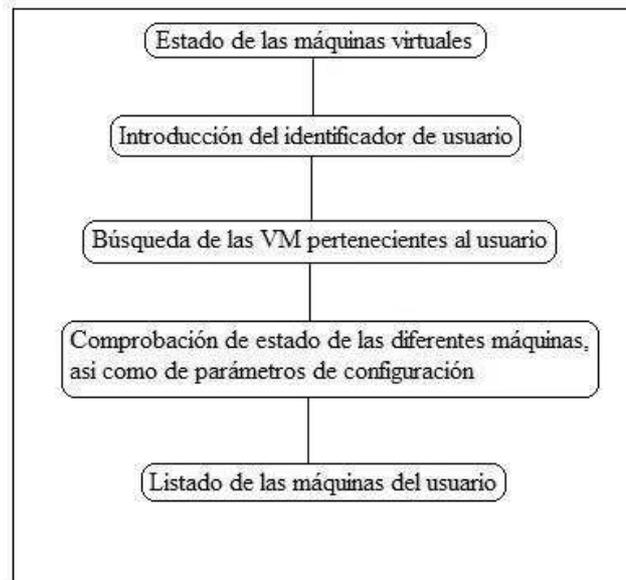


Figura 4.5: Diagrama del apartado de estado de las máquinas virtuales.

### 4.3. Características de la aplicación

Con la aplicación desarrollada en este proyecto podemos crear con facilidad máquinas virtuales según las necesidades que tenga el usuario. El usuario solo debe tener presentes los requerimientos que necesitará su máquina y elegir un tipo según esas necesidades. Las máquinas de cada usuario están identificadas por el niu<sup>4</sup> correspondiente a este, seguido del tipo de sistema operativo de la máquina y de la tipología de esta. Por lo que, para que un usuario pueda encender sus máquinas, modificarlas o ver el estado en que se encuentran, tan sólo deberá introducir su niu y a partir de ahí, la aplicación le mostrará únicamente las máquinas virtuales de su propiedad. En el estado actual en el que se encuentra la aplicación, *solo se nos permite la clonación de una imagen de un servidor GNU/Linux*, por motivos de espacio en la máquina en la que se encuentra instalada.

Durante la creación de las imágenes, se dota a las máquinas virtuales, de una configuración por defecto dependiendo del tipo de sistema operativo elegido y de la función a la que están destinadas. De esta pre-configuración, posteriormente, el usuario podrá si lo ve necesario, modificar algunos de los parámetros como tamaño de la memoria ram o la memoria de video utilizada.

Dentro de la pre-configuración por defecto de las máquinas, cabe destacar la diferencia que se ha hecho entre las máquinas destinadas a ser servidor Web y las que no tienen este fin. Para las máquinas que no sean servidor Web, la configuración de red se realiza mediante NAT<sup>5</sup>. En cambio, para los servidores la conexión se realizará mediante la interfaz del anfitrión [20] por la necesidad que tienen estas máquinas de poseer una dirección IP válida para enrutar el contenido Web que contengan.

Es una aplicación multiplataforma, ya que no importa el sistema operativo que utilicemos ni en nuestro ordenador, ni en nuestra máquina virtual. El proto-

---

<sup>4</sup>Es un número de identificación de los miembros de la comunidad de la UAB. Se ha elegido este por ser un identificador único, pero se podría elegir otro como por ejemplo número de DNI

<sup>5</sup>NAT (Network Address Translation) es un mecanismo utilizado por routers IP para intercambiar paquetes entre dos redes que se asignan mutuamente direcciones incompatibles. Consiste en convertir en tiempo real las direcciones utilizadas en los paquetes transportados. Su uso más común es permitir utilizar direcciones privadas y aún así proveer conectividad con el resto de Internet.

colo RDP, mediante el cual realizamos la conexión, está soportado por múltiples sistemas operativos.

Otro de los puntos interesantes de la aplicación es el poder ejecutar diferentes aplicaciones de forma totalmente independiente. Esta es una de las características de las máquinas virtuales, que nos proporcionan un entorno totalmente aislado, de esta forma podemos ejecutar varios servidores Web en una única computadora, mediante varias VM, sin problemas.

## 4.4. Acceso a la aplicación

Para acceder a la aplicación debemos encontrarnos dentro de la red interna de la UAB, y la podemos realizar mediante la siguiente dirección Web (en la figura 4.6 tenemos un detalle de la sección de creación de nuestra aplicación):

`http://158.109.201.235/index.php`



Figura 4.6: Detalle de la sección de creación de máquinas virtuales.

Una vez creada la máquina virtual, se podrá tener acceso remoto mediante el escritorio remoto que nos proporciona Windows, o bien mediante rdesktop<sup>6</sup> si estamos utilizando una distribución GNU/Linux, en la figura 4.7 tenemos un detalle de nuestra máquina virtual en funcionamiento utilizando rdesktop. Cabe recordar que solo se permiten conexiones dentro de la red de la UAB y que el puerto asignado a la máquina virtual debe estar abierto.

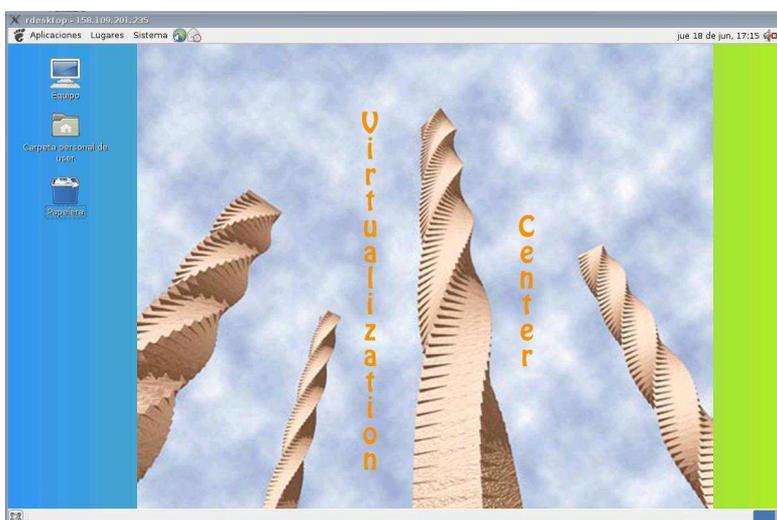


Figura 4.7: Máquina virtual utilizando rdesktop.

## 4.5. Posibles vías de desarrollo

A la hora de futuras mejoras el método de utilización y creación de nuevas máquinas sería el mismo, pudiendo ofrecer un gran abanico de posibilidades. Teniendo como base esta aplicación se nos presentan diversas vías de desarrollo que

---

<sup>6</sup>rdesktop (Remote Desktop Protocol Client) es un cliente del protocolo RDP de Microsoft, escrito en C (trabaja directamente sobre las Xs, X11) y publicado bajo licencia GPL (GNU General Public License) que funciona sobre Linux, BSD y otros sistemas derivados de UNIX.

podrían ser explotadas en un futuro. Algunas de estas vías de desarrollo podrían ser las siguientes:

- Mejora de la seguridad de la aplicación: si bien, en el estado actual en que se encuentra la aplicación la autenticación la tenemos desactivada, se podría crear un grupo de usuarios concretos pudiendo utilizar otro tipo de autenticación como la externa. Esto nos llevaría a la siguiente vía de explotación de la aplicación.
- Creación de imágenes predefinidas para el uso de alumnos: de todos es sabido que en ciertos momentos se requiere el uso de software privativo durante el transcurso de los estudios superiores, lo que hace difícil que el alumnado tenga una versión del programario en casa. A partir de esto, y creando un número de imágenes, se podría facilitar el acceso a estas máquinas con todo el programario necesario para la utilización de este, sin necesidad de estar físicamente en el centro educativo.
- Creación y gestión de *backups*: dar la posibilidad a los usuarios de crear de forma remota copias de seguridad de sus máquinas virtuales, así como editar comentarios sobre ellas, borrar las que ya no sean necesarias, etc.

## 4.6. Conclusiones

En los apartados anteriores, hemos visto tanto el software como parte del diseño utilizado durante el transcurso de la creación de nuestra aplicación. Pero gran parte del peso de la aplicación, por no decir toda, se centra en la gran capacidad que nos aporta la solución de virtualización elegida (VirtualBox), a la hora de poder gestionar prácticamente todo con sus instrucciones mediante terminal.

Hemos visto como es posible crear de una forma relativamente rápida (en apenas unos minutos) y sencilla una máquina virtual sin tener conocimientos previos, gracias al uso de esta aplicación. También se ha comprobado, durante las pruebas realizadas, que es posible disponer de varios servidores Web totalmente independientes dentro de la máquina anfitriona.

Con todo esto, podemos llegar a concluir que la virtualización nos puede permitir una mejor distribución de los recursos económicos (con la reducción en equipos o en consumo eléctrico, por poner un par de ejemplos). También podría ser nos de utilidad creando máquinas con software específico, disponibles para que los usuarios pudieran hacer libre uso, así como a la hora de testear programas sin poner en riesgo la estabilidad de la máquina anfitriona.



# Capítulo 5

## Conclusiones

El campo de la virtualización nos ofrece una nueva forma de concebir el uso que podemos dar a nuestros computadores. Nuestras máquinas dejan de ser algo estático, desde el punto de funcionalidad del sistema operativo y visto como que solo podemos tener un sistema operativo ejecutándose en nuestra máquina, a algo más dinámico. Nos ofrece el dinamismo de poder estar ejecutando de forma paralela diferentes sistemas operativos; utilizar software que no podríamos utilizar con un sistema operativo, pero sí con el instalado en una de nuestras máquinas virtuales; nos brinda la posibilidad de testear sin consecuencias graves para nuestro sistema, etc.

Pero el estudio realizado, con las pruebas hechas sobre el procesador, memoria y tráfico de red, nos ha mostrado que actualmente las máquinas virtuales no nos aportan un beneficio en la explotación intensiva de uno de estos recursos de nuestros computadores. En los tres casos el anfitrión ha conseguido realizar las tareas asignadas en el menor tiempo. Esto nos lleva a afirmar, que en la actualidad, la mejor utilización que podemos darle a la virtualización es a través de otros usos como el que se ha conseguido con la aplicación propuesta en este trabajo, con la creación de diferentes máquinas virtuales para diferentes usuarios.

La virtualización, por tanto, nos aporta la posibilidad de manejar varias máquinas virtuales en un único computador, ofreciéndonos una reducción en costes de maquinaria y por consiguiente, de consumo eléctrico. Este tipo de funcionalidad,

además, ofrece una herramienta muy potente a los administradores de sistemas. Por ejemplo, para la gestión del sistema, poniendo como ejemplo la gestión de servidores Web, tan solo deberían preocuparse de la correcta creación de la imagen que los debería albergar y de la correcta creación de las conexiones puente, para ofrecerles una ip válida. El resto, clonación de la imagen, encendido/apagado de las máquinas, etc., se puede realizar fácilmente mediante el juego de instrucciones que nos ofrecen estas herramientas, en nuestro caso VirtualBox. Nuestra aplicación, consigue realizar todas estas tareas de una forma totalmente transparente al usuario.

Como posibles vías de desarrollo tenemos: el mejorar el sistema de seguridad de la aplicación confrontando los usuarios que deseen dar uso de las máquinas virtuales con los usuarios existentes en nuestro sistema; la posibilidad de crear un sistema de imágenes con el programario necesario en cada caso, para que sea utilizado por alumnos y/o docentes desde sus respectivos hogares; la creación y gestión de copias de seguridad de las máquinas virtuales de los usuarios; etc.

Es posible que en pocos años la virtualización sea una técnica muy utilizada en todos los ámbitos. Con este proyecto hemos visto algunos de sus puntos fuertes así como los débiles y que usos son más beneficiosos para sacar un mayor rendimiento a nuestros equipos utilizando la virtualización.

# Bibliografía

- [1] Wikipedia [En línea] <[http://es.wikipedia.org/wiki/Máquina\\_virtual](http://es.wikipedia.org/wiki/Máquina_virtual)>.
- [2] [En línea] <<http://www.boo.net/~jasonp/qs.html>>[Consulta 29 Enero 2009].
- [3] [En línea] <<http://virtualoverhead.blogspot.com/2008/01/qu-es-la-virtualizacion.html>>[Consulta 4 Junio 2009].
- [4] [En línea] <[http://www.virtualizacion.com/?page\\_id=7](http://www.virtualizacion.com/?page_id=7)>[Consulta 4 Junio de 2009].
- [5] [En línea] <<http://vmware.com/es/overview/history.html>>[Consulta 19 Marzo 2009].
- [6] [En línea] <<http://vmware.com/es/overview/why.html>>[Consulta Febrero 2009].
- [7] Gerald J. Popek and Robert P. Goldberg (1974). «Formal Requirements for Virtualizable Third Generation Architectures », 1974, ISSN:0001-0782.
- [8] [En línea] <<http://es.wikipedia.org/wiki/CTSS>>[Consulta 15 Mayo 2009].
- [9] [En línea] <<http://www.idg.es/computerworld/Microsoft-reduce-diferencias-con-VMware-/seccion-ti/noticia-72689>>[Consulta 4 Junio 2009].

- [10] [En línea] <<http://www.vmware.com/products/lcm/>>[Consulta 18 Mayo 2009].
- [11] Smith, Daniel E.; Nair, Ravi (2005). «The Architecture of Virtual Machines», 2005, ISSN:0018-9162.
- [12] White paper, VMware Infrastructure architecture overview.
- [13] White paper, Security design of the VMware infrastructure 3 architecture.
- [14] White paper, Oracle VM with IBM System x.
- [15] QEMU [En línea] <<http://www.nongnu.org/qemu/userdoc.html>>[Consulta 10 Noviembre 2008].
- [16] Jeanna N. Matthews, Eli M. Dow, Todd Deshane, Wenjin Hu, Jeremy Bongio, Patrick F. Wilbur, Brendan Johnson (2008 by Prentice Hall). «Running Xen: A Hands-On Guide to the Art of Virtualization» [En línea] The XEN hypervisor <<http://www.informit.com/articles/article.aspx?p=1187966>>[Consulta 10 Noviembre 2008].
- [17] Wikipedia [En línea] <<http://es.wikipedia.org/wiki/Virtualización>>[Consulta 2 Noviembre 2008].
- [18] [En línea] <[http://www.ainfor.es/pdf/Ponencia\\_Virtualizacion\\_HP.pdf](http://www.ainfor.es/pdf/Ponencia_Virtualizacion_HP.pdf)>[Consulta 4 Junio 2009].
- [19] Sun xVM VirtualBox, (2004-2008). «User Manual».
- [20] [En línea] <<http://tecnologiaxa.blogspot.com/2008/10/instalar-y-configurar-virtualbox-ii.html>>[Consulta 20 Mayo 2009].

---

Firmado: Alberto Garcia Calahorro  
Bellaterra, Junio de 2009

## **Resum**

La virtualització és una tècnica que té els seus orígens a la dècada dels seixanta i que s'ha anat desenvolupant d'una manera desigual fins als nostres dies. Amb ella es pot simular diverses màquines dintre d'un mateix computador.

Durant el transcurs d'aquest projecte, es veurà l'estat actual d'aquesta tecnologia, es farà un estudi de rendiment de diferents components del hardware del nostre computador i per últim, es mostrarà l'aplicació que s'ha desenvolupat per treure profit de les característiques de les màquines virtuals, mitjançant la creació, gestió i modificació a partir d'un interface Web.

## **Resumen**

La virtualización es una técnica que tiene sus orígenes en la década de los sesenta y que se ha ido desarrollando de manera desigual hasta nuestra época. Con ella se puede simular varias máquinas dentro de un mismo computador.

Durante el transcurso de este proyecto, se verá el estado actual de esta tecnología, se realizará un estudio de rendimiento sobre diferentes componentes del hardware de nuestro computador y por último, mostraremos la aplicación que se ha desarrollado para sacar provecho de las características de las máquinas virtuales, mediante la creación, gestión y modificación a través de un interface Web.

## **Abstract**

The (PC) virtualization has its origin in the 60's decade and it has been developed in an irregular way until today. With this technique we can emulate some PC inside only one.

In this project we will see the state of the art of this technology due to a study with several VM and over different hardware parameters. After this study we will explain the design and development of an application to create, modify and administer the VM with a Web interface.