Universitat Oberta de Catalunya

TFC – Plataforma GNU/Linux

VIRTUALIZACIÓN SOBRE PLATAFORMA OPENSOURCE

ALTERNATIVAS A LAS SOLUCIONES COMERCIALES

MEMORIA

Carlos Miguel Martín Hernández

ITIS

Miquel Angel Senar Rosell

2010/2011-2

Licencia: Creative Commons.

Agradecimientos

Este trabajo de final de carrera está especialmente a mi mujer Ana Angélica. Sin su fervoroso apoyo y constancia, el sueño de alcanzar definitivamente el final de esta ingeniería no se habría hecho realidad.

Dedicado también a mi tutor Miguel Ángel Senar, que me ha guiado a lo largo de este tortuoso semestre, y gracias a su apoyo y comentarios, este trabajo ha cogido forma.

Finalmente, no quiero olvidar a mi familia. Mis padres y mi hermana. Sin su continuo soporte no habría llegado estar en este punto.

Resumen

La virtualización es uno campo relacionado con las nuevas tecnologías de la información que más ha avanzado en los últimos tiempos. El considerable aumento de potencia y rendimiento de los sistemas actuales ha llevado a intentar maximizar el uso de los mismos. Esto, sumado a la necesidad por parte de las empresas de minimizar el espacio ocupado por su infraestructura, costes de consumo eléctrico y contratos de soporte, ha dado el empuje definitivo para el despegue de esta tecnología.

El presente proyecto pretende realizar un estudio de viabilidad de las soluciones de virtualización basadas en software Open Source, con la finalidad de esclarecer su grado de madurez para poder ser implantadas en entornos empresariales complejos y de gran capacidad. Se pretende con ello lograr una reducción de costes globales minimizando lo máximo posible cualquier merma en rendimiento o funcionalidades. Para ello, se parte de un cliente con una infraestructura virtual basada en un hipervisor comercial y se analizarán las opciones Open Source que permitan mantener el mayor número de características del producto ya instalado. Se introducirán las tecnologías Open Source de virtualización y aquellas que tengan mayor potencial para cubrir las necesidades del cliente se probarán y analizarán. De tal manera, que se obtenga un diseño final robusto y de fácil gestión para los integrantes de los departamentos tecnológicos de las empresas.

A partir de este diseño, se elaborará una maqueta funcional y se probarán todas las características que hayan sido requisito del cliente.

Índice de Contenidos e ilustraciones

Agradecimientos 3

Resumen 4

Índice de Contenidos e ilustraciones 5

Índice de Ilustraciones 6

Índice de Tablas 7

1 Memoria 8

1.1 Capítulo I - Introducción 8

1.1.1 Justificación y contexto 8

1.1.2 Objetivos 8

1.1.3 Enfoque y método de trabajo 8

1.1.4 Planificación 9

1.1.5 Productos Obtenidos 13

1.1.6 Otros Capítulos 13

1.2 Capítulo II – Virtualización Open Source. Migración Entornos Comerciales 14

1.2.1 Introducción a la Virtualización 14

1.2.2 Historia de la Virtualización 16

1.2.3 Virtual Machine Monitor (VMM) 16

1.2.4 Métodos de Virtualización 17

1.2.5 Infraestructura Virtual Open Source 20

1.2.6 Descripción de la Arquitectura del Cliente 24

1.2.7 Diseño de la Arquitectura Hardware 25

1.2.8 Pruebas de los hipervisores open source 26

1.2.9 Pruebas de Rendimiento 29

1.2.10 Conclusión final 38

2 Glosario 41

3 Bibliografía y Referencia 43

Bibliografía 43

Referencias 43

4 Anexos 45

4.1 Preparación de los Servidores Físicos 45

4.2 Instalación ProxMox VE 46

4.2.1 Instalación de los nodos 46
4.2.2 Configuración del clúster 47
4.2.3 Configuración del disco de la SAN 48
4.2.4 Repositorio de ISO de Sistemas Operativos 50
4.2.5 Creación de Máquinas Virtuales 51
4.2.6 Clonado de Servidores 54
4.3 Instalación de Xen Cloud Platform (XCP) 55
4.3.1 Instalación de los nodos 55
4.3.2 Creación del Clúster (Pool) 59
4.3.3 Creación del almacenamiento compartido para máquinas virtuales 59
4.3.4 Habilitar el servicio de Alta Disponibilidad 62
4.3.5 Entornos de Gestión 63
4.3.6 Creación de Máquinas Virtuales 64
4.3.7 Clonado de Máquinas Virtuales 66
4.4 Plantilla KickStart de CentOS 67
4.5 Instalación/Configuración de Phoronix Test Suite 68
4.6 Resultados completos de las pruebas de rendimiento 70
4.6.1 pts/linux-system 70
4.6.2 pts/network 75
4.6.3 pts/disk 76
4.6.4 pts/java 80
4.6.5 pts/phpbench 81
4.6.6 pts/memory 81

Índice de Ilustraciones

Ilustración 1 dos Máquinas Virtuales corriendo sobre un VMM que se ejecuta sobre el hardware físico. 17
Ilustración 2 dos máquinas virtuales sobre un hipervisor montado directamente sobre el hardware. 18
Ilustración 3 dos máquinas virtuales haciendo uso de drivers paravirtualizados para acceder a los dispositivos de E/S. 19
Ilustración 4 dos Máquinas Virtuales corriendo sobre un VMM que se ejecuta sobre un Sistema Operativo Anfitrión que es quien tiene acceso al hardware físico. 20
Ilustración 5 esquema de arquitectura del hipervisor Xen. 22
Ilustración 6 esquema de la arquitectura del hipervisor KVM. 23
Ilustración 7 Diseño hardware de la plataforma de virtualización propuesta. 26

Ilustración 8 esquema lógico del funcionamiento de Xen Cloud Platform 27
Ilustración 9 esquema lógico de las pruebas de rendimiento de carga escalable 36

Índice de Tablas

Tabla 1 Hardware de la plataforma de virtualización del cliente 25
Tabla 2 Características de las máquinas del análisis de rendimiento 30
Tabla 3 resultado de las Pruebas usando el perfil pts/cpu 31
Tabla 4 resultado de las pruebas ejecutando el perfil pts/multicore 32
Tabla 5 resultado de las pruebas ejecutando el perfil pts/memory 32
Tabla 6 resultado de las pruebas ejecutando el perfil pts/disk 33
Tabla 7 resultado de las pruebas ejecutando el perfil pts/network 33
Tabla 8 resultado de las pruebas ejecutando el perfil pts/linux-system 34
Tabla 9 resultado de las pruebas ejecutando el perfil pts/java 34
Tabla 10 resultado de las pruebas ejecutando el perfil pts/phpbench 35
Tabla 11 resultado de las pruebas ejecutando el perfil pts/linux-system con OpenVZ..... 35
Tabla 12 resultado de las pruebas ejecutando el perfil pts/apache combinadas..... 37
Tabla 13 resultado de las pruebas ejecutando el perfil pts/pgbench combinadas 37
Tabla 14 resultado de las pruebas ejecutando el perfil pts/x264 combinadas 37

1 Memoria

1.1 Capítulo I - Introducción

Las empresas modernas de un tamaño considerable y las instituciones públicas han invertido a lo largo de los años gran cantidad de recursos en virtualización. Este hecho, es incuestionable si se desea continuar con un crecimiento operacional en la base de servicios ofrecidos dentro de unos valores contenidos en cuanto a espacio ocupado, consumo eléctrico o coste del parque tecnológico de sus Centros de Producción de Datos.

La virtualización permite resolver estos problemas, aportando una solución que aprovecha al máximo las actuales arquitecturas de procesadores.

1.1.1 Justificación y contexto

Estas grandes inversiones suelen venir aparejadas con un considerable gasto anual en licencias de soporte y mantenimiento. Este aspecto es crítico tanto en empresas, como en la administración pública, donde esta bolsa económica repercute directamente en los presupuestos, que se han de invertir en costes de propiedad física e intelectual, en vez de en servicios a la sociedad.

Además, el actual marco de crisis económica impulsa a reducir estos grandes presupuestos. En este marco, es donde el software libre puede entrar y marcar la diferencia.

Cuando una empresa o institución no puede hacer frente al gasto anual en cuanto a soporte y mantenimiento, es más deseable realizar una inversión en software libre invirtiendo en personal que lo mantenga; que continuar utilizando software propietario desfasado y sin actualización.

Este proyecto, pretende dar una solución dentro del ámbito de las infraestructuras de virtualización.

1.1.2 Objetivos

El objetivo no es otro que investigar el estado actual de los proyectos de virtualización Open Source y aplicarlo para dar una solución de software libre a una gran empresa o ente público que tenga una gran base virtual basada en software propietario.

Mediante este estudio, se intentará averiguar si el nivel del software Open Source está a la altura de las soluciones comerciales más comunes. Particularmente, se tomará como base del estudio uno de los productos comerciales con más presencia en el mercado, y que además posee instalado el cliente en la actualidad. Este producto es el hipervisor de la empresa VMWare, Virtual Infraestructure ESX 3.5.

1.1.3 Enfoque y método de trabajo

El proyecto requiere un enfoque de investigación y análisis de una infraestructura ya creada para poder migrarla sin pérdida de funcionalidades importantes a un entorno de software libre.

Para ello, se ha realiza primero un trabajo de documentación sobre los distintos tipos de virtualización existentes que ha derivado en una introducción de base teórica en el presente proyecto.

Seguidamente, se analizarán distintas soluciones de virtualización basadas en software libre y fundamentadas bajo el abanico del sistema operativo GNU/Linux.

Se realizarán diversas instalaciones de diferentes productos para evaluar sus características y si pueden o no ser capaces de soportar la infraestructura virtual del cliente. Para lo cual, se ha hecho acopio de un estudio de la instalación cliente; el software comercial que utilizan; el número y tipo de máquinas virtuales que se alojan; e incluso las tareas de gestión realizadas por el departamento IT.

Con toda la información, se planteará un diseño que será implementado en hardware similar o igual al instalado. Se elaborarán una serie de pruebas/benchmarks para medir el rendimiento del diseño propuesto. De tal manera que nos permita establecer si habrá pérdida de funcionalidad o rendimiento.

1.1.4 Planificación

Para el desarrollo efectivo de este proyecto, se seguirá una estructura de fases dependiente de los hitos concretos que representan los entregables que se han de aportar. Así pues, definimos las siguientes particiones del proyecto:

- Obtención de Requisitos y Planificación (PAC1)

  En esta primera fase, se realizarán las tareas necesarias para definir y plantear los objetivos que se quieren cubrir dentro del proyecto. Además, se tomarán de la empresa, los requisitos iniciales que se desean implementar.

  Se realizará la temporización inicial de todo el proyecto en base a los datos obtenidos previamente, mediante la presentación de una planificación inicial de tareas y la presentación de un Plan de Trabajo (PAC1).

- Estudio de las soluciones y diseño final (PAC2)

  Como segunda fase, se procederá a documentar un estudio de las posibles tecnologías dentro del marco de los sistemas GNU\Linux que puedan dar soluciones al objetivo planteado.

  Comenzando con una breve introducción a la virtualización, sus orígenes y beneficios. Se ahondará en profundidad en las plataformas Xen y KVM para poder diseñar la arquitectura que se asemeje lo más posible a la instalación comercial de la empresa.

  Como resultado final de esta fase del proyecto, se habrá optado por alguna de las tecnologías y se habrá diseñado una arquitectura hardware que permita resolver las necesidades del cliente.

- Implementación (PAC3)

A partir de la base de las fases anteriores, se comenzará a implementar la solución final. Para lo cual, se realizará una instalación de un clúster de servidores físicos que conformen nuestra infraestructura virtual sobre plataforma GNU/Linux.

Posteriormente, se ejecutarán pruebas para verificar el nivel de cumplimiento de los requisitos definidos. Validando, por tanto la instalación realizada.

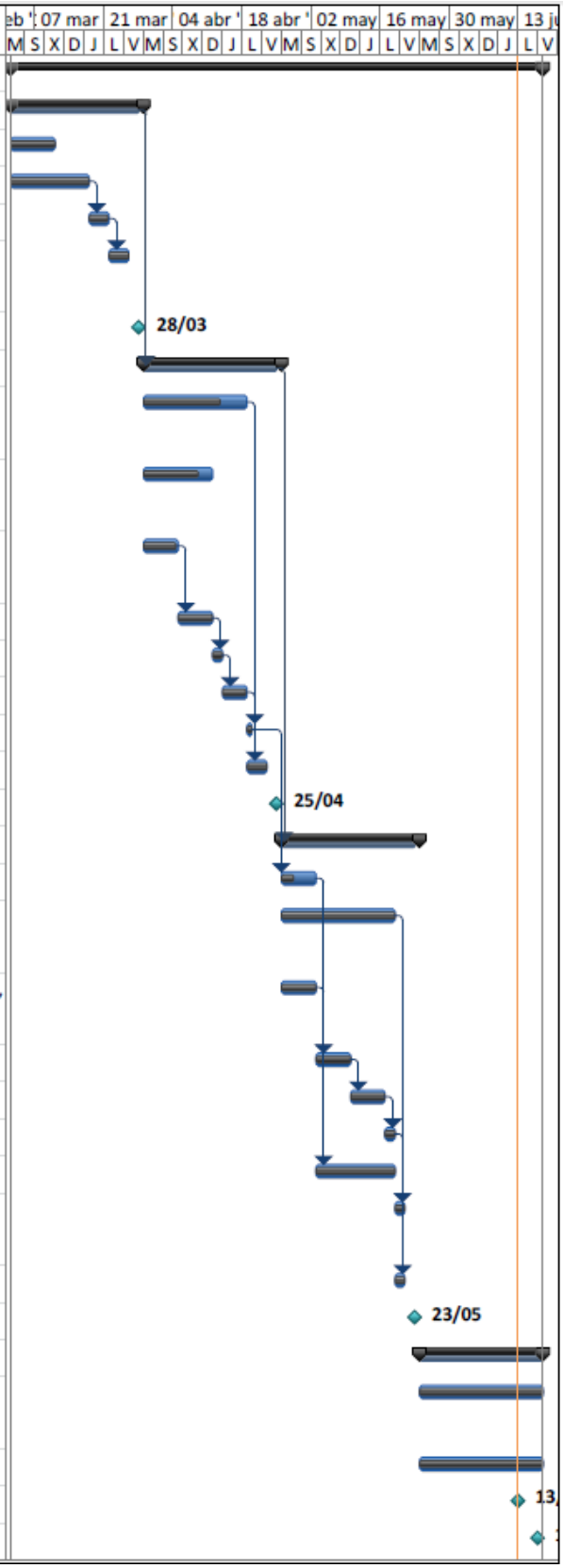
A continuación, se comprobará mediante una serie de pruebas de rendimiento entre las distintas tecnologías evaluadas, si las soluciones open source pueden ofrecer el mismo nivel de servicio que entornos comerciales consolidados en el mercado.

Terminados los pasos descritos, se realizarán una serie de conclusiones sobre la experiencia.

Planificación y Diagrama de Gantt

Nombre de tarea	Duración	Comienzo	Fin	Prede.
TFC GNU/Linux	**78 días**	**mié 02/03/11**	**vie 17/06/11**	
PAC1 (Toma de requisitos y Plan de Trabajo)	**19 días**	**mié 02/03/11**	**lun 28/03/11**	
Revisión de la documentación del TFC	7 días	mié 02/03/11	jue 10/03/11	
Análisis.	12 días	mié 02/03/11	jue 17/03/11	
Obtención de requisitos de la empresa.	2 días	vie 18/03/11	lun 21/03/11	4
Elaboración de la Documentación PAC1 (Descripción y Planificación)	4 días	mar 22/03/11	vie 25/03/11	5
Entregable. PAC1.	0 días	lun 28/03/11	lun 28/03/11	
PAC2 (Análisis y Diseño)	**20 días**	**mar 29/03/11**	**lun 25/04/11**	**2**
Documentación. Preparación de Primera versión Memoria (Índice/Intro)	15 días	mar 29/03/11	lun 18/04/11	
Investigación. Soluciones Virtualización GNU/Linux	5 días	mar 29/03/11	lun 04/04/11	
Estudio y Análisis de las Soluciones.	5 días	mar 05/04/11	lun 11/04/11	10
Revisión de los objetivos de la empresa.	2 días	mar 12/04/11	mié 13/04/11	11
Selección de la Plataforma final.	1 día?	jue 14/04/11	jue 14/04/11	12
Diseño de la Arquitectura.	5 días	vie 15/04/11	jue 21/04/11	13
Preparación entregable PAC2.	4 días	mar 19/04/11	vie 22/04/11	9
Entregable. PAC2.	0 días	lun 25/04/11	lun 25/04/11	
PAC3 (Implementación)	**20 días**	**mar 26/04/11**	**lun 23/05/11**	**8**
Documentación. Continuación trabajo sobre la memoria.	17 días	mar 26/04/11	mié 18/05/11	
Preparación de hardware y requisitos (Red, Disco)	5 días	mar 26/04/11	lun 02/05/11	
Instalación Plataforma Base	5 días	mar 03/05/11	lun 09/05/11	21
Clústerización de la Solución	5 días	mar 10/05/11	lun 16/05/11	22
Ejecución de Pruebas	2 días	mar 17/05/11	mié 18/05/11	23
Benchmarking	12 días	mar 03/05/11	mié 18/05/11	19
Elaboración de conclusiones.	2 días	jue 19/05/11	vie 20/05/11	24
Preparación Entregable PAC3.	2 días	jue 19/05/11	vie 20/05/11	20
Entregable. PAC3.	0 días	lun 23/05/11	lun 23/05/11	
Memoria y Presentación Virtual	**19 días**	**mar 24/05/11**	**vie 17/06/11**	
Documentación. Redacción final de la Memoria.	19 días	mar 24/05/11	vie 17/06/11	
Elaboración de la Presentación	19 días	mar 24/05/11	vie 17/06/11	
Entregable. Memoria.	0 días	lun 13/06/11	lun 13/06/11	
Entregable. Presentación.	0 días	vie 17/06/11	vie 17/06/11	

Id	Nombre de tarea
1	**TFC GNU/Linux**
2	**PAC1 (Toma de requisitos y Plan de Trabajo)**
3	Revisión de la documentación del TFC
4	Análisis.
5	Obtención de requisitos de la empresa.
6	Elaboración de la Documentación PAC1 (Descripción y Planificación)
7	Entregable. PAC1.
8	**PAC2 (Análisis y Diseño)**
9	Documentación. Preparación de Primera versión Memoria (Índice/Intro)
10	Investigación. Documentación. Historia. Tipos y Plataformas de virtualización
11	Investigación. Soluciones Virtualización GNU/Linux
12	Estudio y Analisis de las Soluciones.
13	Revisión de los objetivos de la empresa.
14	Solucitud y Preparación de maquetas.
15	Selección de la Plataforma final.
16	Preparación entregable PAC2.
17	Entregable. PAC2.
18	**PAC3 (Implementación)**
19	Diseño de la Arquitectura.
20	Documentación. Continuación trabajo sobre la memoria.
21	Preparación de hardware y requisitos (Red, Disco)
22	Instalación Plataforma Base
23	Clusterización de la Solución
24	Ejecución de Pruebas
25	Benchmarking
26	Elaboración de las conclusiones
27	Preparación Entregable PAC3.
28	Entregable. PAC3.
29	**Memoria y Presentación Virtual**
30	Documentación. Redacción final de la Memoria.
31	Elaboración de la Presentación
32	Entregable. Memoria.
33	Entregable. Presentación.



1.1.5 Productos Obtenidos

El resultado final de este proyecto intentará ofrecer la mejor de las soluciones open source dentro del campo de la virtualización que sea capaz de acomodar una gran infraestructura de servicio basados en máquinas virtuales. Todo ello, sin perder funcionalidades, ofreciendo el máximo posible de rendimiento y otorgando al diseño de la máxima capacidad posible de disponibilidad de los servicios ofrecidos.

Por lo tanto, se ofrecerá un diseño hardware/software basado en la solución de las analizadas que tenga todas estas cualidades.

1.1.6 Otros Capítulos

A lo largo del siguiente capítulo, se desarrollará la parte descriptiva y funcional de este proyecto. Se comenzará con una introducción sobre los conceptos generales que existen detrás de las tecnologías de virtualización. Para ello, describiremos el significado de la misma e integraremos las últimas tendencias en un marco de tiempo, describiendo la evolución de esta tecnología hasta la actualidad.

Se hará especial énfasis en la descripción de los diferentes tipos de virtualización y las clasificaciones de los mismos más comúnmente utilizadas. Esto permitirá obtener una base de información y vocabulario necesarios para encasillar cada solución existente dentro de un tipo u otro.

Para continuar, se describirán las principales tecnologías haciendo balance entre sus características principales y comparándolas entre ellas. Seguidamente, se tomarán distintos productos que ejemplaricen estas tecnologías y se verá cómo se adaptan a un entorno moderno de producción.

Se tomarán estos productos como base de un análisis de rendimiento de características más detallado. Para lo cual, se ha elaborado un marco de pruebas que contrastará cada una de las soluciones contra la versión de software comercial que el cliente posee en producción.

Finalmente, con la información obtenida, se dictarán una serie de conclusiones y se propondrá un diseño al cliente que le permita migrar su infraestructura virtual de software propietario a un entorno open source.

1.2 Capítulo II – Virtualización Open Source. Alternativas a las soluciones Comerciales

1.2.1 Introducción a la Virtualización

Recientemente, una de las tecnologías de la sociedad de la información de las que más se habla y que más ha evolucionado es la virtualización. Años atrás, la virtualización no era tomada en cuenta como una alternativa real al momento de instalar servidores y otros equipos de producción en la mayoría de los Centros de Datos, debido mayormente a que era una tecnología poco probada, costosa, ineficiente, o por el "miedo a lo nuevo". Sin embargo, hoy en día la virtualización se ha posicionado en el mercado de la informática como una opción económica y efectiva al momento de diseñar, ampliar, y actualizar tecnología de Centros de Datos, y en muchos casos si no se elige la virtualización, se estaría perdiendo dinero y/o la implementación podría ser menos efectiva.

Esta reflexión nos lleva a preguntarnos ¿Qué es la Virtualización? Según la Wikipedia la podemos definir como:

"En Informática, **virtualización** se refiere a la abstracción de los recursos de una computadora, llamada Hypervisor o VMM (Virtual Machine Monitor) que crea una capa de abstracción entre el hardware de la máquina física (host) y el sistema operativo de la máquina virtual (virtual machine, guest), siendo un medio para crear una versión virtual de un dispositivo o recurso, como un servidor, un dispositivo de almacenamiento, una red o incluso un sistema operativo, donde se divide el recurso en uno o más entornos de ejecución."

Ampliando un poco esta definición, básicamente podemos definir la virtualización como una tecnología que te permite instalar y configurar múltiples computadoras y/o servidores completamente independientes (conocidas como "virtual machines" o "máquinas virtuales") en una sola "caja" física, ya sea una computadora, servidor, "appliance", etc. A pesar de que estas máquinas virtuales comparten todos los recursos de un mismo "hardware", cada una trabaja de manera totalmente independiente (con su propio sistema operativo, aplicaciones, configuraciones, etc.). En otras palabras, en lugar de utilizar 5 servidores físicos, cada uno de ellos corriendo una aplicación que solo utiliza el 10% de los recursos de su servidor; se puede instalar 5 máquinas virtuales, cada una con su propia aplicación y configuraciones específicas, en un solo servidor y utilizar el 50-60% de los recursos del mismo. Cabe señalar que cada una de estas máquinas virtuales, con la debida configuración, deberá funcionar exactamente igual que un servidor físico. Los objetivos finales de la adopción de esta tecnología serán:

- Índices de utilización más alto. Antes de la virtualización, los índices de utilización del servidor y almacenamiento en los centros de datos de la empresa rondaban menos del 50% (de hecho, del 10% al 15% de los índices de utilización fueron los más comunes). A través de la virtualización, las cargas de trabajo pueden ser encapsuladas y transferidas a los sistemas inactivos o sin uso — lo cual significa que los sistemas existentes pueden ser consolidados, así que las compras de capacidad adicional del servidor pueden ser retrasadas o evitadas.

- Consolidación de Recursos. La virtualización permite la consolidación de múltiples recursos de TI. Más allá de la consolidación de almacenamiento, la virtualización proporciona una oportunidad para consolidar la arquitectura de sistemas, infraestructura de aplicación, datos y base de datos, interfaces, redes, escritorios, e incluso procesos de negocios, resultando en ahorros de costo y mayor eficiencia.
- Uso/costo menor energía. La electricidad requerida para que funcionen los centros de datos de clase empresarial ya no está disponible en suministros ilimitados, y el costo está en una espiral ascendente. Por cada dólar gastado en un servidor hardware, un dólar adicional es gastado en energía (incluyendo el costo de los servidores en función y los enfriadores). Utilizando virtualización para consolidar hace posible cortar el consumo total de energía y ahorrar dinero de una manera significativa.
- Ahorros de espacio. La extensión del servidor permanece como un serio problema en la mayoría de los centros de datos empresariales, pero la expansión del centro de datos no es siempre una opción, con los costos de construcción promediando miles de dólares por pie cuadrado. La virtualización puede aliviar la tensión mediante la consolidación de muchos sistemas virtuales en menos sistemas físicos.
- Recuperación de desastre/continuidad del negocio. La virtualización puede incrementar la disponibilidad de los índices del nivel de servicio en general y proporcionar nuevas opciones de soluciones para la recuperación de desastre.
- Costos de operación reducidos. La virtualización puede cambiar el radio de servicio a administración reducir la carga total de trabajo administrativo, y cortar el total de costos de operación.

Y además obteniendo las siguientes ventajas:

- Rápida incorporación de nuevos recursos para los servidores virtualizados.
- Reducción de los costes de espacio y consumo necesario de forma proporcional al índice de consolidación logrado (Estimación media 10:1).
- Administración global centralizada y simplificada.
- Nos permite gestionar nuestro CPD como un pool de recursos o agrupación de toda la capacidad de procesamiento, memoria, red y almacenamiento disponible en nuestra infraestructura
- Mejora en los procesos de clonación y copia de sistemas: Mayor facilidad para la creación de entornos de test que permiten poner en marcha nuevas aplicaciones sin impactar a la producción, agilizando el proceso de las pruebas.
- Aislamiento: un fallo general de sistema de una máquina virtual no afecta al resto de máquinas virtuales.
- Mejora de TCO y ROI.
- No sólo aporta el beneficio directo en la reducción del hardware necesario, sino también los costes asociados.
- Reduce los tiempos de parada.
- Migración en caliente de máquinas virtuales (sin pérdida de servicio) de un servidor físico a otro, eliminando la necesidad de paradas planificadas por mantenimiento de los servidores físicos.

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

1.2.2 Historia de la Virtualización

Contrario de lo que la mayoría piensa, el tema de la vitalización no es nada nuevo. Durante la década de los 60 los equipos de informática de muchas empresas y entidades tenían un problema similar: contaban con super-computadoras o "mainframes" de alto rendimiento que deseaban "particionar lógicamente", o utilizar para múltiples tareas simultaneas (lo que hoy se conoce como "multitasking", trabajar más de una aplicación o proceso simultáneamente). Es por esto que IBM desarrolló un método para crear múltiples "particiones lógicas" (similar a lo que se denomina hoy como "máquinas virtuales") las cuales trabajaban independientemente una de las otras, y cada una utilizando los recursos provistos por el "mainframe".

Ya para la década de los 80 y con la llegada de las relativamente económicas maquinas x86, comenzó una nueva era de micro computadoras, aplicaciones cliente-servidor, y "computación distribuida"; en donde los enormes y potentes "mainframes" con mil y una tareas y utilidades en una sola caja gigantesca se comenzaron a cambiar por relativamente pequeños servidores y computadoras personales de arquitectura x86, con "una caja diferente para cada uso", lo que se convirtió rápidamente en el estándar de la industria. Debido a esto, una vez más, el tema de la virtualización vuelve a quedar prácticamente en el olvido. No siendo hasta finales de la década de los 90 que gracias al alto desarrollo del hardware se vuelve a caer en un predicamento similar al acaecido en los años 60: el hardware existente es altamente eficiente, y utilizar cada "caja" para una sola aplicación sería un desperdicio de recursos, espacio, energía y dinero; y tampoco es conveniente asignarle múltiples usos o instalar varias aplicaciones en un solo servidor convencional, por más de una razón (ej. estas aplicaciones podrían ser conflictivas entre sí, o podrían requerir diferentes configuraciones e inclusive diferentes sistemas operativos, o tener diferentes requerimientos de seguridad, entre otras variables que podrían causar problemas al ejecutar estas funciones simultáneamente). Es por esto que vuelve a resurgir la idea de dividir el hardware, de manera tal que funcione como múltiples servidores independientes pero compartiendo los recursos de un mismo servidor físico. A raíz de aquí nace lo que hoy se conoce comúnmente como "Virtualización".

1.2.3 Virtual Machine Monitor (VMM)

Un Monitor de Máquina Virtual (VMM), también conocido como Hipervisor, es un conjunto se software o hardware que se ejecuta sobre los dispositivos físicos del servidor anfitrión. Representa la capa de abstracción entre el hardware físico y las máquinas virtuales que se ejecutan sobre una capa por encima. Todos los servidores huésped son controlados y monitorizados por el VMM. Es el encargado de proveer las herramientas para que los administradores gestionen las máquinas virtuales.

Una máquina virtual suele tener una CPU virtual. El VMM mapea las CPU virtuales de todas las máquinas virtuales que se estén ejecutando en el servidor con las CPU físicas del equipo

anfitrión. De hecho, normalmente habrá más máquinas virtuales ejecutándose en el servidor, que CPU físicas tenga este, lo que requiere de algún tipo de planificador. Por lo tanto, un VMM usa una especie de mecanismo de planificación para compartir las CPU físicas con cada CPU virtual.

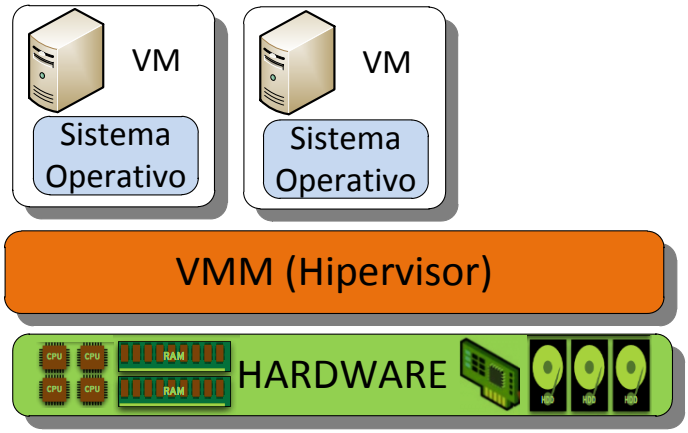


Ilustración dos Máquinas Virtuales corriendo sobre un VMM que se ejecuta sobre el hardware físico.

Además, un VMM ha de hacerse cargo de la gestión de memoria. Se encarga de mapear una cierta cantidad de memoria física en las máquinas virtuales y ha de hacerse cargo de la posible fragmentación y swapping. Dado que algunas máquinas virtuales requerirán de mayor o menor cantidad de memoria que otras, se deberá de poder ajustar de forma dinámica mediante las herramientas de gestión.

De forma general, las máquinas virtuales no tienen acceso al hardware físico y ni siquiera son conscientes de las características del mismo. Según la arquitectura del VMM y cuando se desee, se podrán dejar pasar directamente algunas características del hardware físico. Sin embargo, el escenario más común es el VMM quien provee de los dispositivos virtuales de E/S, como: tarjetas de red, discos duros y lectores de CD. Puesto que los VMM tienen una misma configuración de hardware virtual para las diferentes máquinas virtuales, la posibilidad de migrar las máquinas virtuales entre servidores anfitrión (host), ejecutando el mismo sistema VMM no es una tarea complicada. Además, permite estandarizar las instalaciones y la gestión de drivers.

1.2.4 Métodos de Virtualización

Dentro del amplio campo de la virtualización, nos centraremos específicamente en aquella que atañe a la virtualización de servidores. Y en esta especialización podemos destacar diferentes métodos. Básicamente podemos considerar 4 tipos de virtualización: emulación, virtualización completa (Full Virtualization), paravirtualización (Paravirtualization) y virtualización sobre sistema operativo.

1.2.4.1 Emulación

La emulación se basa en crear máquinas virtuales que emulan el hardware de una o varias plataformas hardware distintas. Este tipo de virtualización es la más costosa y la menos eficiente, ya que obliga a simular completamente el comportamiento de la plataforma

hardware a emular e implica también que cada instrucción que se ejecute en estas plataformas sea traducida al hardware real.

Sin embargo, la emulación tiene características interesantes, como poder ejecutar un sistema operativo diseñado para una plataforma concreta sobre otra plataforma, sin tener que modificarlo, o en el desarrollo de firmware para dispositivos hardware, donde se pueden comenzar estos desarrollos sin tener que esperar a tener disponible el hardware real.

Uno de los ejemplos más destacados de la actualidad es QEMU. QEMU, entre otras cosas, permite emular diferentes plataformas Hardware como x86, x86-64, PowerPC, SPARC o MIPS. Así pues, podríamos tener dentro de un servidor linux varios equipos x86 o PowerPC, corriendo diferentes versiones de Linux.

La emulación en si, no corre sobre un Hipervisor como tal pues suele ser un producto cerrado especializado en la simulación completa de un sistema sobre otro.

1.2.4.2 Virtualización completa

Con este término se denominan aquellas soluciones que permiten ejecutar sistemas operativos huésped (Guest), sin tener que modificarlos, sobre un sistema anfitrión (Host), utilizando en medio un Hipervisor o Virtual Machine Monitor que permite compartir el hardware real. Esta capa intermedia es la encargada de monitorizar los sistemas huésped con el fin de capturar determinadas instrucciones protegidas de acceso al hardware, que no pueden realizar de forma nativa al no tener acceso directo a él.

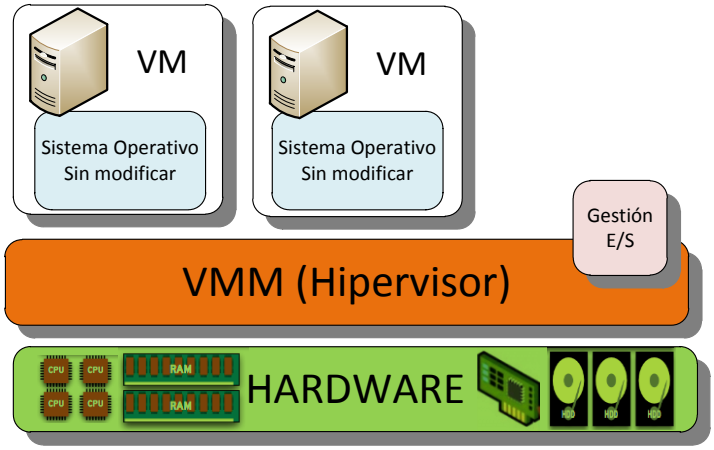


Ilustración dos máquinas virtuales sobre un hipervisor montado directamente sobre el hardware.

Su principal ventaja es que los sistemas operativos pueden ejecutarse sin ninguna modificación sobre la plataforma, aunque como inconveniente frente a la emulación, el sistema operativo debe estar soportado en la arquitectura virtualizada.

En lo que respecta al rendimiento, éste es significativamente mayor que en la emulación, pero menor que en una plataforma nativa, debido a la monitorización y la mediación del hipervisor. Sin embargo, recientes incorporaciones técnicas en las plataformas x86 hechas por Intel y AMD, como son Intel VT y AMD-V, han permitido que soluciones basadas en la virtualización completa se acerquen prácticamente al rendimiento nativo.

Un par de ejemplos significativos de este tipo de VMM son VMWare ESX, KVM o Xen con extensiones de Intel o AMD.

Hay que tener en cuenta también que la virtualización completa no se refiere a todo el conjunto de hardware disponible en un equipo, sino a sus componentes principales, básicamente el procesador y memoria. De esta forma, otros periféricos como tarjetas gráficas, de red o de sonido, no se virtualizan. Las máquinas huésped no disponen de los mismos dispositivos que el anfitrión, sino de otros virtuales genéricos.

Este tipo de VMM son conocidos como *de Hipervisores de Tipo 1*.

1.2.4.3 Paravirtualización

La paravirtualización surgió como una forma de mejorar la eficiencia de las máquinas virtuales y acercarlo al rendimiento nativo. Para ello se basa en que los sistemas virtualizados (huésped) deben estar basados en sistemas operativos especialmente modificados para ejecutarse sobre un Hipervisor. De esta forma no es necesario que éste monitorice todas las instrucciones, sino que los sistemas operativos huésped y anfitrión colaboran en la tarea.

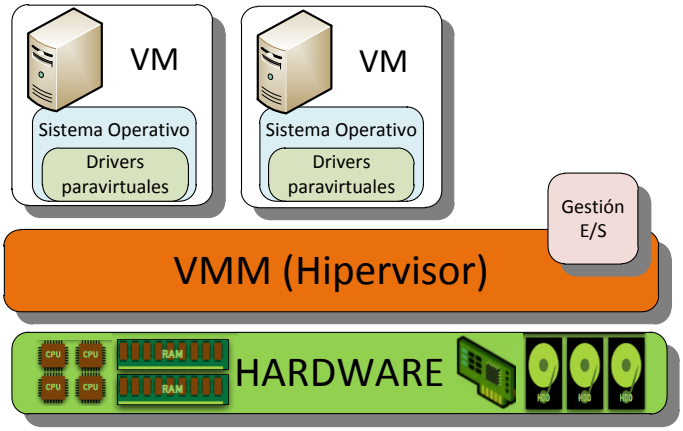


Ilustración dos máquinas virtuales haciendo uso de drivers paravirtualizados para acceder a los dispositivos de E/S.

Este tipo de virtualización está muy optimizado, ofreciendo normalmente más rendimiento que la virtualización completa. Se ha de tener en cuenta que las máquinas virtuales son conscientes de lo que son y tienen acceso casi directo a los recursos hardware, siempre bajo el control del hipervisor que ofrece calidad de servicio y seguridad.

Uno de los componentes más destacados de esta familia es XEN (Oracle VM está basado en XEN). Permite paravirtualización utilizando sistemas operativos modificados, y virtualización completa sobre procesadores con tecnología Intel-VT o AMD-V.

1.2.4.4 Virtualización sobre sistema operativo

Uno de los tipos de virtualización más comunes es aquella que se realiza sobre un sistema operativo. Sobre este, se ejecuta el software hipervisor, que se encarga de traducir todas las llamadas del sistema virtualizado al sistema operativo sobre el que está corriendo.

De forma general, este tipo de hipervisores no tiene acceso al hardware del anfitrión y es el sistema operativo que está bajo él quien acceso a la E/S y le ofrece estos servicios al hipervisor.

Este tipo de arquitectura no ofrece tanto rendimiento como como la virtualización completa o la paravirtualización. Sin embargo, posibilita las posibilidades de configuración son muy flexibles y permite combinar varios sistemas operativos en un único equipo.

Es un tipo de virtualización muy extendida en entorno de escritorio, y los ejemplos más representativos son: VMWARE Server, Parallels, etc.

Los que se acaban de describir se suelen denominar *Hipervisores de Tipo 2*.

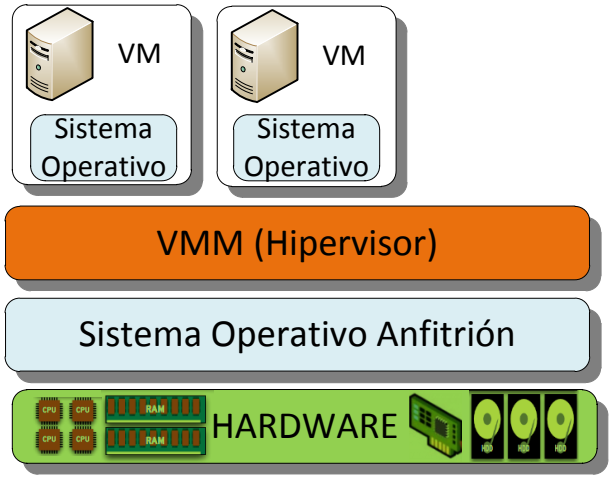


Ilustración dos Máquinas Virtuales corriendo sobre un VMM que se ejecuta sobre un Sistema Operativo Anfitrión que es quien tiene acceso al hardware físico.

1.2.5 Infraestructura Virtual Open Source

En el entorno Open Source han aparecido diversas tendencias y modelos de virtualización para dar solución a los problemas de la arquitectura x86 y en general, de la saturación de los centros de datos actuales. En el presente proyecto nos centraremos en aquellas que han obtenido mayor crédito y penetración en las empresas.

Estos son los hipervisores Xen y KVM. Por lo tanto, se realizará una introducción a estas tecnologías describiendo su evolución y características. Además, se introducirá el producto que se probará de cada una de estas tecnologías, con la intención de encontrar aquel que encaje mejor con los requisitos del cliente y en general, con cualquier entorno empresarial de tamaño medio o grande.

1.2.5.1 Xen

1.2.5.1.1 Historia

Xen fue creado en el año 2003 en el laboratorio de computación de la Universidad de Cambridge bajo lo que se conoce como el proyecto Xen Hipervisor liderado por Ian Pratt. Algunos de los miembros más destacados del proyecto son Keir Fraser, Steven Hand y Christian Limpach. Este mismo equipo fundó XenSource en el 2004 conjuntamente con Nick Gault y Simon Crosby, que aportaron su experiencia como empresarios en Silicon Valley, con la intención de transformar el Hipervisor Xen de una herramienta de investigación a un producto competitivo para las empresas de computación.

Fue a lo largo del 2004 cuando aparece la primera versión de Xen la 1.0 seguida en corto plazo por la versión 2.0. Y como parte de su visión corporativa, mantuvieron el producto como una solución Open Source. Con la versión 2.0 comienza una amplia adopción por parte de diversas compañías, destacando RedHat, Novell y Sun. Todos ellos añadieron las capacidades de virtualización del Hipervisor Xen a sus sistemas operativos. Gracias a estas colaboraciones, y a la comunidad de desarrolladores Open Source, el desarrollo se aceleró hasta que en 2005 aparece la versión 3.0.

A partir del año 2006 la empresa Citrix se hace con los derechos de este hipervisor y comienza a liderar desarrollo y evolución pero sin abandonar su ideología Open Source. Para ello, han aportado financiación a la fundación Xen y han integrado el hipervisor en un producto comercial denominado XenServer que ofrece una solución robusta, estable y con características alineadas con las necesidades de las empresas.

Por otro lado, el núcleo del desarrollo del hipervisor bajo el paraguas de Citrix ha evolucionado y aumentado sus características y funcionalidades hasta la actual versión 4.1.

1.2.5.1.2 Arquitectura

Por lo general, las máquinas virtuales necesitan imitar el hardware que un sistema necesita. El inconveniente es que el hardware emulado es mucho más lento que el real. Xen adopta una perspectiva diferente, puesto que restringe la emulación al número mínimo posible de partes. Para lograrlo, Xen utiliza la paravirtualización. Ésta es una técnica que presenta máquinas virtuales similares, aunque no idénticas al hardware subyacente. Por lo tanto, los sistemas operativos del host y del invitado se adaptan al nivel del núcleo. El espacio de usuario permanece sin cambios. Xen controla el hardware con un hipervisor y un invitado controlador (también denominado "dominio-0") que proporcionan los dispositivos virtuales de bloque y de red necesarios. Los sistemas invitados utilizan estos dispositivos para ejecutar el sistema y realizar la conexión con otros invitados o con la red local. Cuando varias máquinas físicas que ejecutan Xen se configuran de tal forma que los dispositivos virtuales de bloque y de red se encuentran disponibles, también resulta posible migrar un sistema invitado de un elemento de hardware a otro durante su ejecución. Originariamente, Xen se desarrolló para funcionar hasta con 100 sistemas invitados en un solo equipo; no obstante, este número presenta una fuerte dependencia de los requisitos del sistema de los sistemas invitados en ejecución (sobre todo del uso de la memoria).

Para limitar el uso de la CPU, el hipervisor de Xen ofrece tres planificadores distintos. El planificador también puede cambiarse durante la ejecución del sistema invitado, con lo que se posibilita el cambio de prioridad del sistema invitado en ejecución. En un nivel más alto, la migración de un invitado también puede utilizarse para ajustar los recursos disponibles de la CPU.

El sistema de virtualización Xen también presenta algunas desventajas relacionadas con el hardware admitido. Principalmente por la necesidad de compilar los drivers para el dom0 que es quien los gestiona, lo cual, en caso de controladores propietarios, puede dar problemas.

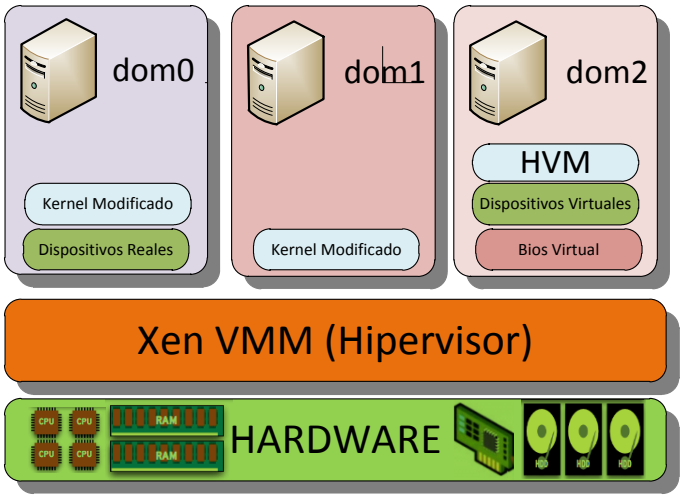


Ilustración esquema de arquitectura del hipervisor Xen.

1.2.5.2 KVM

1.2.5.2.1 Historia

KVM (Kernel Virtual Machine) fue desarrollado inicialmente por la empresa israelita Qumranet a lo largo del año 2006. Sin embargo, no llenaría a tener la repercusión que ha adquirido hoy en día hasta la adquisición en septiembre del 2008 por parte de RedHat. Esta empresa ha destacado al producto como la nueva generación dentro de la tecnología de virtualización. Hoy en día es el VMM por defecto de las distribuciones Enterprise de RedHat desde la versión RedHat Enterprise Linux (RHEL) 5.4.

Qumranet liberó el código de KVM a la comunidad Open Source. Y hoy, compañías del calado de IBM, Intel y AMD están en la lista de contribuidores al proyecto. Desde la versión 2.6.20 del núcleo de Linux, KVM se parte integral del mismo quedando sus cualidades disponibles desde esta versión a las actuales. Con esto, KVM se aprovecha del conocimiento del grupo de desarrollo del sistema operativo Linux, y dado que está en continuo avance, KVM también se aprovecha de las mejoras de rendimiento, funcionalidades y nuevos drivers.

En resumen, KVM es un sistema de virtualización que utiliza el mecanismo de Virtualización Completa para ejecutar las Máquinas Virtuales (VM). Además, posee una huella muy pequeña de código fuente, lo que hace que su mantenimiento hoy en día sea muy sencillo. KVM soporta principalmente hardware de arquitectura x86, pero se está trabajando para ampliar su disponibilidad a otros entornos.

1.2.5.2.2 Arquitectura

KVM utiliza el kernel de linux como VMM dado que este sistema operativo posee todos los mecanismos que un VMM necesita para operar varias Máquinas Virtuales. Como la gestión de dispositivos, planificador de CPU, gestor de memoria, etc. Tal y como se avanzó en el apartado específico sobre las características de los VMM. Por lo tanto, los desarrolladores de KVM no han de reinventar la rueda y solo han de añadir algunos componentes específicos para soportar la virtualización.

KVM se ha implementado como un módulo del kernel de Linux y que puede ser cargado por este para extender las capacidades de virtualización del sistema operativo. Y por lo tanto, convertir al servidor Linux en un VMM.

En un entorno Linux normal cada proceso se ejecuta o bien en modo usuario, o bien en modo kernel. KVM añade un tercer modo, el modo invitado. Sin embargo, para ello se apoya en una CPU con capacidades de virtualización y con soporte bien de Intel VT o de AMD-V. Un proceso en modo invitado tiene su propio modo usuario y modo kernel. Y por tanto, es capaz de ejecutar un sistema operativo. Estos procesos representan las Máquinas Virtuales en un host KVM.

En general, distinguimos desde el punto de vista del anfitrión, los siguientes modos de funcionamiento:

- user-mode: E/S cuando el sistema operativo invitado necesite acceso a los dispositivos.
- kernel-mode: encargado del paso a guest-mode y del manejo de las salidas a través de las operaciones de E/S.
- guest-mode: ejecuta el código del invitado/guest, meno aquellas que sean de E/S.

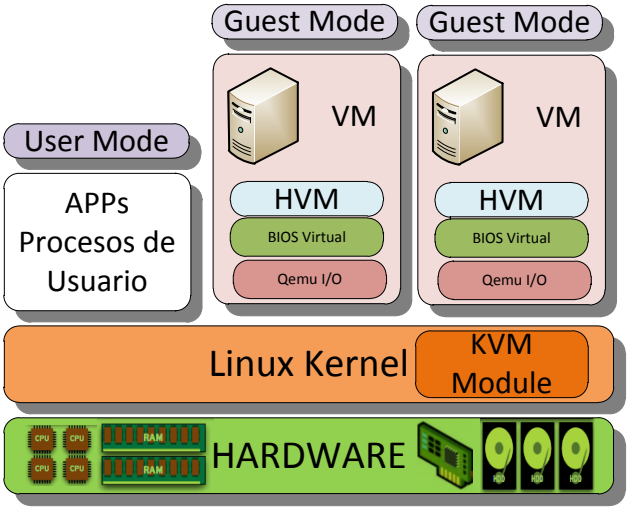


Ilustración esquema de la arquitectura del hipervisor KVM.

1.2.5.3 OpenVZ

Una solución muy particular, que difiere de las dos presentadas anteriormente es OpenVZ. Este nombre, que proviene de Open VirtualiZations, utiliza tecnología de virtualización basada en el nivel del Sistema Operativo. Mediante OpenVZ se nos permite ejecutar múltiples instancias de sistema operativo sobre un servidor físico. Estas instancias son conocidas como contenedores, Virtual Private Servers (VPSs), o Virtual Enviroments (VEs). Es un proceso similar a las FreeBSD jails y las Solaris Zones.

Comparado a tecnologías como la de VMWARE o la paravirtualización de la que hace gala Xen, OpenVZ está limitado tanto en cuanto el sistema operativo anfitrión y el invitado han de ser Linux (aunque los contenedores pueden ser cualquier tipo diferente de distribución Linux). Sin

embargo, la gran ventaja que reclama OpenVZ es el rendimiento. Tomando la página web de OpenVZ como referencia, la pérdida de rendimiento comparada con el servidor físico es de sólo un 1-3%. Métricas que pueden ser ciertas o no según como se hayan realizado las pruebas.

OpenVZ, como la gran mayoría de sistemas de virtualización cuenta con el respaldo de una compañía comercial, en este caso está a cargo de la empresa Parallels, Inc.

1.2.6 Descripción de la Arquitectura del Cliente

Se ha realizado un estudio del entorno virtual del cliente. Para ello, se han analizado tres aspectos fundamentales del entorno. Primero, se ha revisado la base hardware en la que se está ejecutando el entorno virtual. Segundo, se ha revisado el software hipervisor que existe instalado. Se ha realizado una reunión con los administradores para evaluar las características que utilizan y cuáles no. Finalmente, se ha desglosado el volumen de máquinas virtuales que se encuentran instaladas, de tal manera que nos sea posible definir la distribución de sistemas operativos, versiones y valores medios de memoria, disco, etc.

A continuación, se desglosan los resultados de este análisis. A partir del cual podremos definir un diseño y una solución de las que se han probado que se acomode a las necesidades del cliente.

1.2.6.1 Entorno Hardware

La arquitectura virtual del cliente está basada en chasis Blade. La tecnología Blade permite cumplir una de las máximas de la virtualización. Concretamente, la consolidación de servidores en el menor espacio posible. Aunando para ello, una solución de centralizada de alimentación y refrigeración. Y sin desperdiciar potencia.

Cada servidor blade es una delgada "tarjeta" que contiene únicamente microprocesador, memoria y buses. Es decir, no son directamente utilizables ya que no disponen de fuente de alimentación ni tarjetas de comunicaciones.

Estos elementos más voluminosos se desplazan a un chasis que se monta en el bastidor. Cada chasis puede albergar del orden de dieciséis "tarjetas" o servidores blade (según fabricante). El chasis lleva integrados los siguientes elementos, que son compartidos por todos los servidores:

- Fuente de alimentación: redundante y hot-plug.
- Ventiladores o elementos de refrigeración.
- Conmutador de red redundante con el cableado ya hecho, lo que simplifica su instalación.
- Interfaces de almacenamiento. En particular, es habitual el uso de redes SAN (Storage Area Network) de almacenamiento.

Concretamente, el cliente hace uso de Chasis del modelo HP c7000. Estos equipos permiten un total de 16 tarjetas (servidores) del modelo HP BL460c G1. Los cuales cuentan con las siguientes características:

Tabla Hardware de la plataforma de virtualización del cliente

CPU	2	X5255 Intel Quad Core 2.66 Ghz (1333Mhz) L2 8MB
Memoria (GB)	8*2	16GB Internal Memory ECC
HDD (GB RAID1)	2	60GB
Network (Gb)	2	NetXtreme II BCM5708S Gigabit
RAID I/O Device	1	HP Smart Array E200i
HBA I/O Devices	2	Qlogic ISP2432 4Gb Fibre Channel HBA

1.2.6.2 Requisitos Iniciales

La empresa desea mantener la mayor parte de las funcionalidades ofrecidas por la plataforma comercial. Es decir:

- Alta disponibilidad de Red. La solución adoptada ha de ser tolerante a una caída alguno de los enlaces de red.
- Alta disponibilidad de disco. La solución adoptada ha de ser tolerante a una caída alguno de los enlaces de disco.
- Clúster de Hosts Físicos que permita:
    - Soporte de máquinas virtuales Linux y Windows.
    - Soporte de máquinas virtuales x86 y x86-64.
    - Migración en caliente de las máquinas virtuales.
    - Balanceo de carga.
    - Mantenimiento de los Host Físicos.
    - Snapshots de las máquinas virtuales.
    - Asignación de múltiples CPU virtuales.

Otro requisito impuesto por la empresa, es la utilización de distribuciones compatibles con RedHat. Debido, principalmente, a que la experiencia de sus empleados está basada sobre esta versión de Sistema Operativo.

1.2.7 Diseño de la Arquitectura Hardware

Para la ejecución de pruebas y definición del diseño final de la solución al cliente, este nos ha aportado un entorno lo más fiel posible al actual. Sobre este entorno, se han instalado diversos productos de virtualización para poder realizar de la forma más fiel posible el análisis de la opción más óptima que se puede implementar en base a los requisitos iniciales.

El entorno que el cliente nos ha aportado es el siguiente:

Dos servidores Blade pertenecientes a un Chasis HP c7000. Concretamente, el modelo de las tarjetas son BL460c G1 con dos CPU Quad-Core X5355, 16G de RAM y un RAID0 de 60G con discos SAS de 10000 RPMS. Además, con 2 interfaces Gigabit Ethernet y 2 puertos HBA conectados a una SAN de IBM.

En cuanto al almacenamiento, se dispone de un conexionado completo y configurado hacia la SAN del cliente a través de fibra y haciendo uso del protocolo Fiber Channel. Desde la SAN se ofrecen 2 discos o LUN de 150G cada una para almacenar las máquinas virtuales.

La conexión Fiber Channel a la cabina de almacenamiento para acceder al disco compartido por parte de todos los nodos de la solución que se instale se realiza a través de los switches de fibra que poseen los servidores Blade.

La conectividad está a cargo de la infraestructura del propio cliente. Los chasis poseen sus propios switches de comunicaciones que se encadenan a su vez con los corporativos. Aunque en las pruebas no se utilizado, la gestión de la red se realiza a través de VLAN y el protocolo 802.1Q.

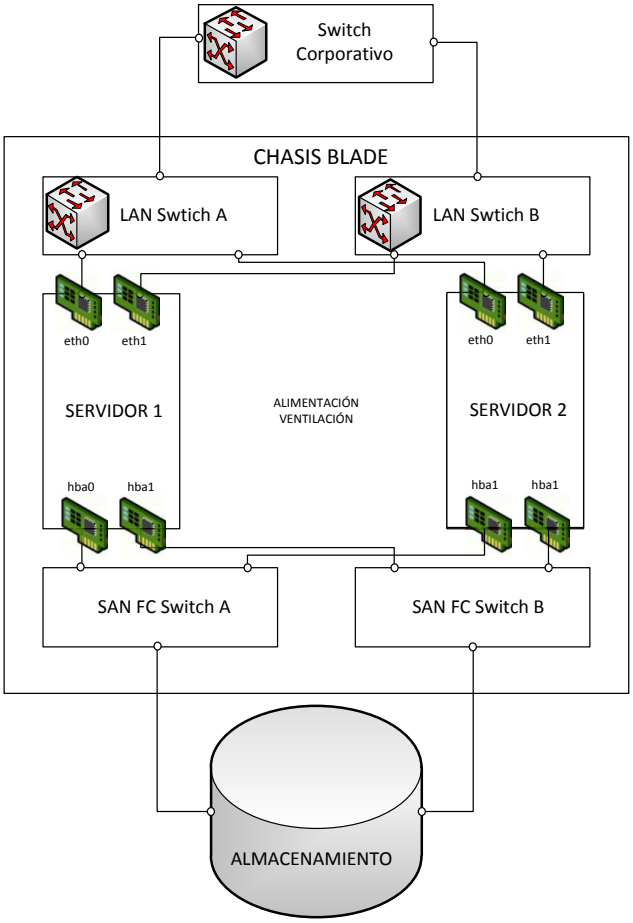


Ilustración Diseño hardware de la plataforma de virtualización propuesta.

1.2.8 Pruebas de los hipervisores open source

Tras un extenso análisis de las tecnologías de virtualización disponibles bajo el ámbito de los desarrollos Open Source. Se ha decidido probar dos de ellos principalmente. En general, muchos han sido descartados por la inmadurez del producto, falta de estabilidad o bien por no cumplir con alguno de los requisitos iniciales.

Centrándonos principalmente en los hipervisores basados en KVM y XEN, destacamos dos propuestas que sobresalen sobre todas las demás.

Los dos productos poseen una serie de características que los asemejan a la solución comercial de VMWARE que el cliente tiene desplegada.

1.2.8.1 Xen Cloud Platform (XCP)

Xen Clound Platform es una plataforma de virtualización de servidores Open Source. Su concepción la hace ideal para la empresa. Basada en el hipervisor Xen, soporta un amplio rango de sistemas operativos huésped entre los que destacan Linux y Windows. Además, acepta una amplia gama de sistemas de red y almacenamiento. Todo ello, en una única y simple imagen ISO lista para ser instalada sobre un servidor físico.

XCP es un desarrollo originado a partir de Citrix XenServer y se encuentra licenciado bajo la licencia GNU General Public License (GPL2) encontrándose disponible sin coste alguno tanto en formato binario como su código fuente.

Su mayor diferencia sobre cualquier otra arquitectura Xen es XAPI. Esta es la interfaz de comunicación de procesos desarrollada sobre el hipervisor para su control y gestión.

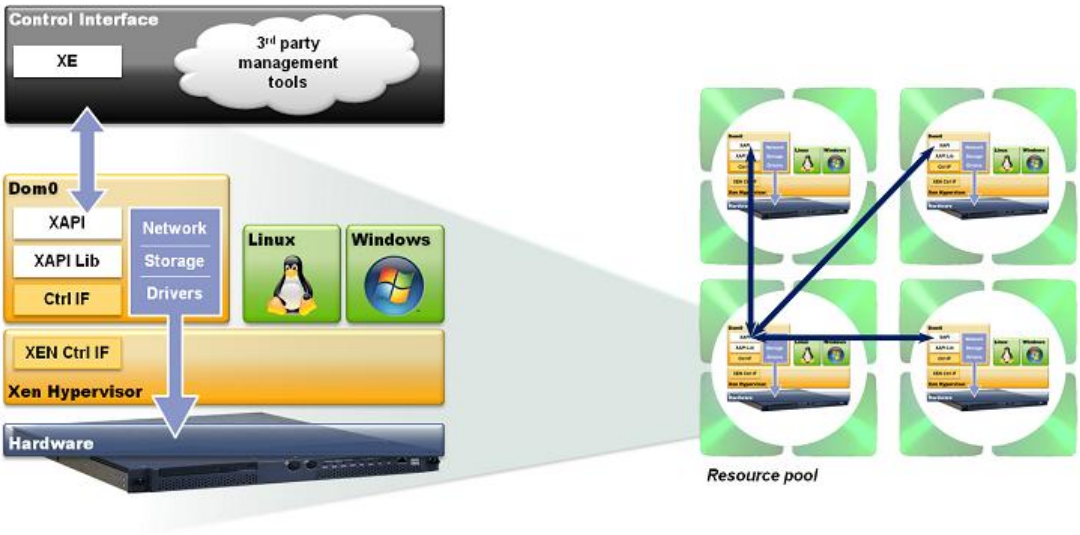


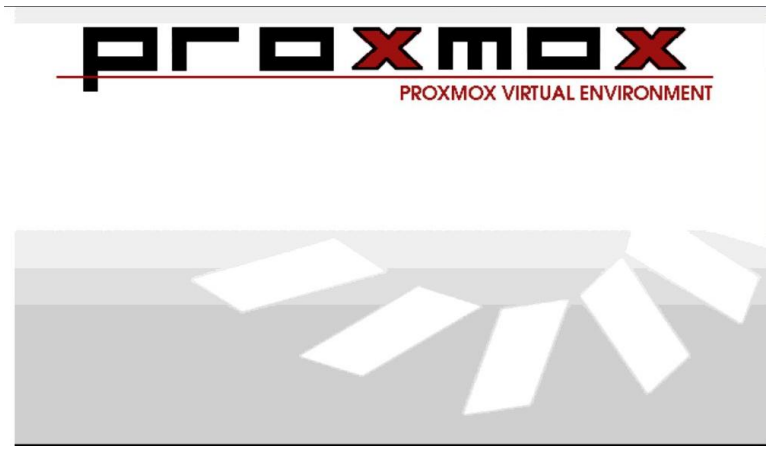
Ilustración esquema lógico del funcionamiento de Xen Cloud Platform

Las características principales que destacan a XCP son:

- XAPI una pila de gestión implementada en Ocaml que permite configurar y controlar los anfitriones Xen y los Pool de recursos. XAPI suplanta en este sistema la XenAPI.

- Interfaces de Control: XCP contiene la utilidad en línea de comando XE para controlar las máquinas virtuales Xen y los Pool de recursos. Además, permite ser controlado por un amplio rango de utilidades de gestión comerciales u open-source.
- Xen Hypervisor: XCP contiene y está basado en el Hipervisor Xen.
- El Dom0 privilegiado, incluye soporte para interfaces de red, almacenamiento y varios drivers: El kernel de Linux del Dom0 de XCP además de incluir drivers para los dispositivos de entrada y salida más comunes, permite su extensibilidad mediante Open vSwitch tal que se pueden crear complicadas infraestructuras en la nube.
- Sistemas Operativos invitados: la distribución XCP soporta por defecto un elevado número de Sistemas Operativos, incluyendo los más comunes Linux y Windows.

1.2.8.2 ProxMox VE



Proxmox VE (Virtual Environment) es una plataforma de virtualización Open Source creada sobre la distribución Debian y que hace uso de las tecnologías de virtualización hardware KVM y de los contenedores software OpenVZ para almacenar y servir las máquinas virtuales.

Esta distribución la mantiene la empresa Proxmox Server Solutions GmbH, afincada en Viena (Austria). Es fundada durante el año 2004 y se especializan en soluciones de correo y anti-spam. El año 2007, y aplicando los conocimientos obtenidos en base a su experiencia en los servicios de correo, crean ProxMox Virtual Environment.

Su principal orientación es la facilidad de uso y el rendimiento. Su desarrollo está centrado con un gran interés en las necesidades empresariales, y para ello, pone a disposición de sus usuarios un gran abanico de appliance virtuales pre-configurados, así como aquellos aportados por la comunidad.

Las principales características que esgrime este producto son:

- Su instalación se puede realizar directamente sobre el hardware del servidor a través de una ISO.
- La versión probada se encuentra basada en la distribución Debian Lenny x86-64 completa.

- No requiere herramientas de gestión. Incluye una consola de gestión vía interfaz web. Utiliza tecnología AJAX y todas sus conexiones van encriptadas (SSL).
- Integra herramientas de copias de seguridad y recuperación.
- Soporta su instalación en clúster. El clúster es gestionado de forma integral desde la misma interfaz web.
- Soporta la migración de servidores virtuales entre los distintos anfitriones del clúster.

Para la elaboración de esta memoria se ha hecho uso de la última versión estable de esta distribución. Concretamente, la versión 1.8. Sin embargo, ya está definida la línea de actualización de las próximas versiones, que según la planificación, tiene prevista su salida para el año 2011. Esta nueva versión contará con las siguientes mejoras:

- Completa re-implementación de la interfaz web con numerosas mejoras: autenticación LDAP o DA, mejorado soporte a cientos de máquinas virtuales, etc.
- Basada en la nueva versión de Debian x86-64 Squeeze.
- Restructuración del clústeres. Soportando entorno multimaster a través de una base de datos compartida. Y con los fundamentos iniciales para el soporte de alta disponibilidad de las máquinas virtuales basadas en KVM.
- Integración con aplicaciones de gestión de terceros.
- Mejoras de rendimiento.
- Y muchas otras.

1.2.9 Pruebas de Rendimiento

Para poder comparar los productos, se han planificado una serie de pruebas de rendimiento basadas en la ejecución de diversos programas que nos permitan emular el uso genérico de las máquinas virtuales que se desean instalar.

Estas pruebas se han basado en la plataforma Phoronix Test Suite.

Phoronix Test Suite es un marco de trabajo de ejecución de pruebas de rendimiento Open Source licenciado bajo GNU GPLv3. Está diseñado para la ejecución de pruebas de rendimiento tanto cualitativas como cuantitativas de forma limpia, reproducible y sencilla de acometer.

Diseñada inicialmente para automatizar las pruebas en Linux, se ha añadido soporte a nuevos sistemas operativos: OpenSolaris, Apple MacOS X, Microsoft Windows y sistemas operativos BSD.

Este sistema ha sido adoptado por numerosas publicaciones para efectuar comparación de rendimiento entre diversas plataformas de hardware.

La comparativa se basa en una instalación estándar del sistema operativo CentOS 5.6 x86-64. Para facilitar esta instalación y asegurar la mayor exactitud entre cada una de ellas, se ha utilizado una plantilla de kickstart (ver anexo). En esta plantilla se han añadido los paquetes necesarios para para solventar las dependencias de software de pruebas Phoronix Test Suite.

Se han desarrollados dos tipos de pruebas diferentes.

- **Pruebas de rendimiento individual**. Se ha evaluado el rendimiento genérico de un único servidor Linux CentOS 5.6.
- **Pruebas de rendimiento escalar**. Se ha evaluado la capacidad de escalabilidad de cada una de las plataformas. La configuración hardware de esta máquina virtual es la misma para los tres entornos.

Tabla Características de las máquinas del análisis de rendimiento

CPU	4
Memoria	2048
HDD	12G
Sistema Operativo	CentOS 5.6

Además, se ha configurado de la forma más eficiente en cada una de las plataformas.

En XCP se ha realizado una instalación del sistema operativo Linux paravirtualizado y se han configurado las xen server tools para mejorar el rendimiento.

En ProxMox VE, se ha instalado el sistema operativo utilizando el virtualizador KVM y se han configurado los drivers VIRTIO tanto de disco como de red para mejorar el rendimiento del sistema.

Finalmente, en VMWARE ESX 3.5 se ha instalado el sistema operativo y configurado las vmware-tools.

1.2.9.1 Pruebas sobre rendimientos individuales

Se ha realizado una instalación igual en los tres entornos del análisis y se ha efectuado un contraste de pruebas de rendimiento. Además, se ha instalado una máquina virtual con Windows 2003 Enterprise SP2 para comparar el rendimiento entre las tres plataformas. De forma excepcional, se ha realizado una prueba del rendimiento de una instalación de CentOS 5.6 sobre OpenVZ en el entorno de ProxMox.

Para la ejecución de estas pruebas se ha utilizado un único servidor anfitrión con una única máquina virtual en ejecución. El objetivo es medir el rendimiento de los componentes virtuales de cada uno de los sistemas de virtualización que se están contrastando. De esta manera, se podrá establecer un marco inicial de comparación de cada una de las tecnologías.

Esta comparativa se ha realizado ejecutando varios conjuntos de pruebas estándar definidas en la plataforma Phoronix Test Suite:

- **Linux-System**. Conjunto agrupado de pruebas para evaluar el rendimiento genérico de un sistema Linux.
- **SMP-Suite**. Pruebas orientadas a estresar el comportamiento de las capacidades de ejecución de procesos simétricos.
- **Disk-Suite**. Conjunto de herramientas para medir el comportamiento de la I/O del sistema.
- **Java-Suite**. Conglomerado de pruebas basadas en java.

- **Memory-Suite**. Grupo de pruebas para comprobar el rendimiento de la RAM del sistema.
- **Multicore-Suite**. Conjunto de pruebas que es capaz de tomar ventaja de un sistema con múltiples procesadores.
- **CPU-Suite**. Este grupo de pruebas está orientado a la medición del rendimiento de la CPU/Procesador del servidor en el que se ejecuta.

Este conjunto de pruebas ha dado como resultado una gran cantidad de información y gráficas sobre el rendimiento global de las tres plataformas analizadas. A continuación, se expone un resumen de los resultados y finalmente, unas conclusiones al respecto de las mismas.

Los cuadros resumen de las pruebas que se muestran a continuación clasifican los resultados obtenidos por cada una de las plataformas. En rojo se destaca el valor menor de la muestra, y en verde su resalta el más favorable. En cada cuadro, se muestra una fila por cada prueba ejecutada.

- **Pruebas de rendimiento de la CPU**. Se muestran dos tablas. La primera de ellas está asociada al perfil pts/cpu y la segunda a pts/multicore. Son un conjunto de pruebas que se especializan en estresar las capacidades del componente CPU del servidor.

Tabla **resultado de las Pruebas usando el perfil pts/cpu**

ptsl Phoronix Test Suite	XCP	KVM	VMWARE
GraphicsMagick	84	80	76
GraphicsMagick	33	31	30
GraphicsMagick	71	65	62
GraphicsMagick	32	31	30
Parallel BZIP2 Compression	16.17	15.69	18.59
7-Zip Compression	5500	7009	6071
LAME MP3 Encoding	34.84	35.66	37.72
FLAC Audio Encoding	12.51	12.60	13.55
x264	38.49	30.65	34.50
FFmpeg	19.16	19.55	21.20
OpenSSL	38.85	38.38	37.00
Gcrypt Library	3496	3490	3617
Himeno Benchmark	180.91	179.60	166.91
PostgreSQL pgbench	1457.34	1359.61	1461.59
Apache Benchmark	7540.83	7746.40	6745.33
C-Ray	90.81	91.48	95.87
POV-Ray	1363	1398	1500
Smallpt	270	273	287
Crafty	747.30	762.73	823.79
TSCP	273542	271802	255600
Timed MAFFT Alignment	36.25	37.17	39.50
NAS Parallel Benchmarks	74.58	74.90	70.62
NAS Parallel Benchmarks	59.50	58.57	54.73
NAS Parallel Benchmarks	5017.40	5029.10	4062.63
NAS Parallel Benchmarks	2395.79	2431.98	1390.05
NAS Parallel Benchmarks	16.50	17.95	12.06
Stream	4857.79	5733.61	3217.12
Stream	4671.37	5455.40	3172.95
Stream	4961.81	5719.90	3447.26
Stream	5021.46	5743.45	3477.85
			OpenBenchmarking.org

Tabla resultado de las pruebas ejecutando el perfil pts/multicore

ptsl Phoronix Test Suite	XCP	KVM	VMWARE
7-Zip Compression	5311	6874	6193
Parallel BZIP2 Compression	15.87	15.62	19.12
x264	37.28	31.62	33.95
OpenSSL	38.53	38.45	37.05
Gcrypt Library	3443	3483	3590
GraphicsMagick	33	31	30
GraphicsMagick	71	65	63
Himeno Benchmark	181.33	178.97	167.17
Bullet Physics Engine	18.30	18.51	19.87
C-Ray	90.65	91.67	95.90
POV-Ray	1362	1401	1505
Smallpt	270	274	287
Crafty	748.20	766.16	824.88
Timed HMMer Search	33.53	34.85	37.24
Timed MAFFT Alignment	36.28	37.16	39.77
Minion	240.81	244.68	267.74
Minion	187.00	190.38	209.79
NAS Parallel Benchmarks	75.27	74.21	71.32
NAS Parallel Benchmarks	59.55	58.19	53.93
NAS Parallel Benchmarks	5064.39	5013.92	4041.79
NAS Parallel Benchmarks	18.26	17.93	12.06
Timed Apache Compilation	45.19	102.63	102.13
OpenBenchmarking.org			

Revisando los resultados de las pruebas, se puedo observar como los resultados favorables (siempre marcados en verde) son claramente favorables al entorno Xen Cloud Platform. Sin embargo, si analizamos detalladamente los datos obtenidos, podemos comprobar como las diferencias entre Xen y KVM no son muy elevadas, y se mueven dentro de un estrecho margen. En cambio, si destaca el bajo rendimiento de la plataforma VMWare ante las condiciones de esta prueba.

- **Memoria**. Para este análisis se utiliza el perfil pts/memory que agrupa la mayoría de las pruebas de rendimiento de memoria del programa Phoronix Test Suite.

Tabla resultado de las pruebas ejecutando el perfil pts/memory

ptsl Phoronix Test Suite	XCP	KVM	VMWARE
RAMspeed SMP	4518.55	3869.55	2882.18
RAMspeed SMP	4547.22	4215.31	2977.06
RAMspeed SMP	4704.46	4554.37	2971.29
RAMspeed SMP	5012.52	3776.23	3217.39
Stream	5267.61	5719.29	3449.85
Stream	5124.35	5732.21	3214.03
Stream	4961.75	5435.73	3172.14
CacheBench	2241.99	2226.55	2099.21
CacheBench	9881.72	9801.82	9191.10
OpenBenchmarking.org			

Nuevamente destaca en estas pruebas los resultados del hipervisor Xen. Pero, nuevamente el margen de mejora sobre KVM es muy bajo. Los resultados de VmWare vuelven a estar muy por debajo de lo esperado por un producto comercial.

- **Disco**. Se ha utilizado un perfil genérico existente en el software de rendimiento llamado pts/disk que agrupa varias pruebas de disco que miden la calidad de este sistema de entrada/salida. Sus principales scripts de prueba están basados en IOZone y Dbench. El primero es un software multiplataforma de realización de pruebas de stress sobre sistemas de ficheros y el segundo, es una versión open-source de netbench diseñada por el proyecto samba para medir el rendimiento de sistemas de ficheros.

Tabla **resultado de las pruebas ejecutando el perfil pts/disk**

ptsh Phoronix Test Suite	XCP	KVM	VMWARE
Gzip Compression	26.40	55.00	25.84
Apache Benchmark	6348.88	7763.29	6732.94
PostgreSQL pgbench	1362.63	1215.45	1415.98
Compile Bench	64.62	77.21	76.60
IOzone	182.74	141.61	210.93
IOzone	46.16	48.50	186.96
Dbench	86.18	99.17	107.31
Dbench	155.70	222.98	294.90
Dbench	201.82	293.18	371.95
Dbench	222.38	296.61	378.92
FS-Mark	107.17	100.40	80.03
Flexible IO Tester	23.10	33.34	28.84
Threaded I/O Tester	260.25	170.74	366.15
Threaded I/O Tester	76.04	52.79	61.85
PostMark	1009	1096	1238
AIO-Stress	165.71	240.91	329.71
Unpacking The Linux Kernel	23.54	22.87	18.43
			OpenBenchmarking.org

Tras la ejecución de este perfil de pruebas, VMWare demuestra que está muy por encima de las dos soluciones Open Source analizadas en cuanto a la maduración de los drivers de acceso a disco. A su vez, las diferencias entre Xen y KVM son mayores y dejan entrever cierta mejoría por parte de la solución KVM.

- **Rendimiento de la Red**. Esta prueba mide el rendimiento del adaptador de loopback ejecutando una prueba de capacidad de transmisión TCP.

Tabla **resultado de las pruebas ejecutando el perfil pts/network**

ptsh Phoronix Test Suite	XCP	KVM	VMWARE
Loopback TCP Network Performance	71.96	191.41	94.40
Difference	1.62x	1.00x	1.51x
Standard Error	1.73	0.83	0.86
Standard Deviation	5.89%	0.75%	1.57%
			OpenBenchmarking.org

Como se observa, XCP nos ofrece el mejor resultado, sin embargo, su diferencia con VMWare nos es tan grande en base a la desviación de las muestras de XCP.

- **Rendimientos de Sistema**. Phoronix Test Suite incorpora una serie de perfiles que ejecutan un conjunto de pruebas con la finalidad de medir el rendimiento general en base al tipo de plataforma. Se han ejecutado tres de estos perfiles, basándonos en las preferencias del cliente en cuanto a sistemas y utilización de las máquinas virtuales.

Tabla **resultado de las pruebas ejecutando el perfil pts/linux-system**

ptsl Phoronix Test Suite	XCP	KVM	VMWARE
LZMA Compression	250.17	257.07	279.92
Parallel BZIP2 Compression	15.10	15.85	19.13
Apache Benchmark	7506.85	7655.98	6703.72
LAME MP3 Encoding	34.88	36.06	38.01
Ogg Encoding	16.23	16.44	17.86
FFmpeg	19.01	19.69	21.38
GnuPG	19.65	19.79	21.32
GMPbench	5411.15		5154.35
C-Ray	90.74	91.50	96.18
POV-Ray	1362	1400	1501
Timed MAFFT Alignment	36.33	37.15	39.80
Timed HMMer Search	33.71	33.90	37.17
Threaded I/O Tester	173.24	193.52	273.04
Threaded I/O Tester	1187.71	936.16	563.63
PostMark	1106	1158	1226
Dbench	168.90	248.05	290.92
GraphicsMagick	86	79	74
GraphicsMagick	33	31	30
OpenSSL	38.88	38.43	36.75
Crafty	748.20	764.39	833.10
Sunflow Rendering System	7.88	7.42	7.73
Sudokut	41.17	43.10	45.66
dcraw	36.05	37.20	39.38
Minion	187.91	189.56	210.16
PyBench	4061	4162	4358
PostgreSQL pgbench	1471.56	1400.39	1438.08
			OpenBenchmarking.org

Tabla **resultado de las pruebas ejecutando el perfil pts/java**

ptsl Phoronix Test Suite	XCP	KVM	VMWARE
Sunflow Rendering System	7.65	7.42	7.76
Bork File Encrypter	47.42	71.40	47.34
Java SciMark	498.17	500.80	470.82
Java SciMark	1056.16	1045.57	973.36
Java SciMark	850.40	844.07	792.68
			OpenBenchmarking.org

Tabla resultado de las pruebas ejecutando el perfil pts/phpbench

Phoronix Test Suite	XCP	KVM	VMWARE
PHPBench	52669	53194	50078
Difference	1.05x	1.06x	1.00x
Standard Error	254.06	482.55	296.23
Standard Deviation	0.84%	1.57%	1.02%

OpenBenchmarking.org

De forma excepcional, se ha realizado un contraste usando la prueba de sistema pts/Linux-System añadiendo a la misma un entorno montando sobre ProxMox mediante el contenedor OpenVZ. A continuación se muestran los resultados.

Tabla resultado de las pruebas ejecutando el perfil pts/linux-system con OpenVZ

Phoronix Test Suite	XCP-TOOLS	VMWARE-TOOLS	KVM-VIRTIO	OPENVZ
LZMA Compression	251.89	279.63	257.20	251.61
Parallel BZIP2 Compression	15.17	18.88	15.68	9.71
Apache Benchmark	8361.42	6870.88	7720.52	7895.90
LAME MP3 Encoding	35.12	38.75	35.52	34.72
Ogg Encoding	16.18	17.95	16.44	16.10
FFmpeg	19.20	21.59	19.71	18.66
GnuPG	19.78	22.18		19.36
GMPbench	5389.90	5079.50		5415.75
C-Ray	91.02	96.54	91.55	47.36
POV-Ray	1369	1525	1402	1365
Timed MAFFT Alignment	36.17	40.25	37.04	36.08
Timed HMMer Search	33.52	37.32	33.47	20.97
Threaded I/O Tester	184.49	249.02	177.71	27.84
Threaded I/O Tester	1004.16	824.69	3300.58	2868.17
PostMark	1086	1263	1136	543
Dbench	166.70	288.48	239.40	17.41
GraphicsMagick	86	73	80	96
GraphicsMagick	34	30	31	49
OpenSSL	38.85	36.68	38.30	38.95
Crafty	748.08	832.98	764.75	749.22
Sunflow Rendering System	7.61	7.83	7.46	4.49
Sudokut	40.79	46.19	42.74	37.90
dcraw	35.86	39.90	37.86	35.40
Minion	186.21	210.12	191.57	
PyBench	4065	4367	4133	4011

OpenBenchmarking.org

Los resultados obtenidos son sorprendentes. La máquina virtual montada sobre OpenVZ es clara vencedora en casi todas las pruebas ejecutadas por este perfil. Su rendimiento se ve reducido únicamente en las pruebas asociadas al disco, donde ya se había visto previamente los problemas con KVM sobre ProxMox. Esta repetición de resultados negativos, se podría achacar al driver qlogic (HBA del sistema) que no sea tan actualizado bajo Debian Lenny. En cualquier caso, las pruebas relacionadas con rendimiento de CPU y aplicaciones dan como claro vencedor a este contenedor de máquinas virtuales.

1.2.9.2 Prueba de escalabilidad

Las plataformas de virtualización están pensadas para asumir una gran carga de trabajo concurrente, en forma de múltiples máquinas virtuales operando al mismo tiempo. Por lo tanto, la forma más óptima de medir este comportamiento es simular en entorno lo más parecido a la situación final en la que se va a encontrar el sistema.

Para ello, se ha construido una prueba combinada con 6 máquinas virtuales ejecutándose de forma simultánea. El objetivo de esta prueba, es simular un entorno de alta carga con diversos perfiles de servidores. Así pues, dos servidores simulando carga web (utilizando apache); dos servidores simulando carga transaccional de base de datos (utilizando postgreSQL); un servidor simulando un proceso de utilización intensiva de CPU (compresión x264) y finalmente, un servidor ocioso con Windows 2003 Enterprise SP2. Este último servidor simula la posibilidad que existe de tener algún servidor sin carga.

Visualmente, el entorno montado en el sistema es el siguiente:

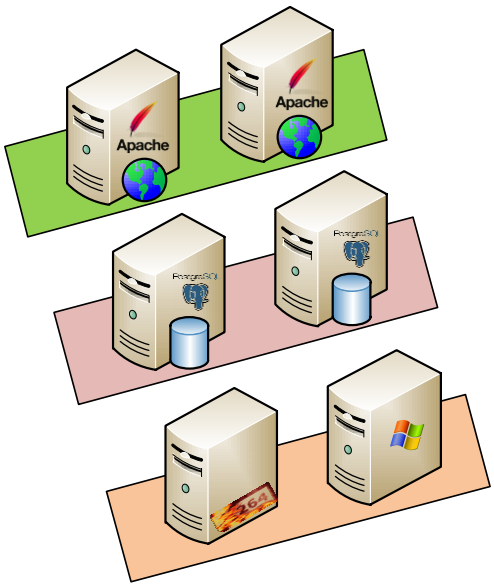


Ilustración esquema lógico de las pruebas de rendimiento de carga escalable

Con todas las máquinas operativas de forma simultánea, se han ejecutado los perfiles de pruebas pts/apache, pts/pgbench y pts/x264. Y una vez finalizados, se realiza un contraste de los resultados. Gráficamente se pueden ver a continuación:

Tabla resultado de las pruebas ejecutando el perfil pts/apache combinadas

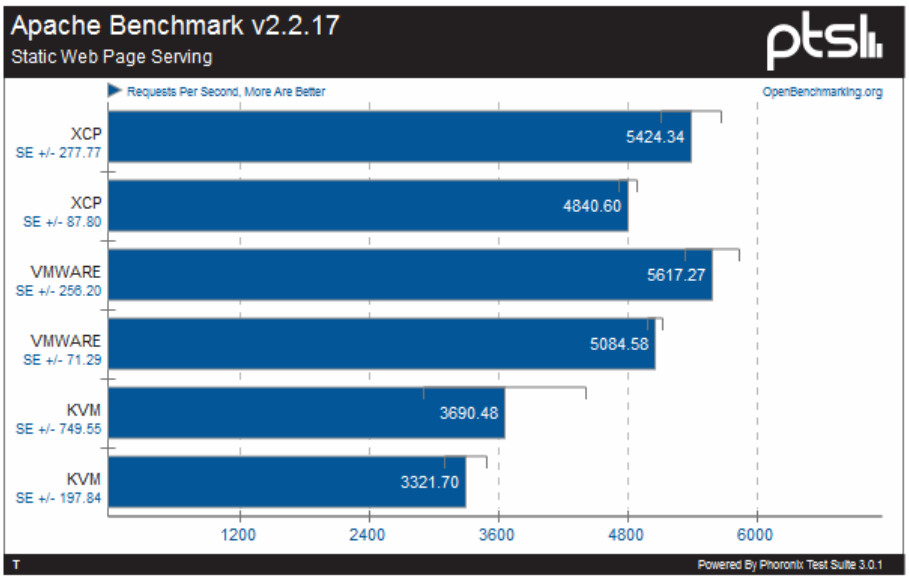


Tabla resultado de las pruebas ejecutando el perfil pts/pgbench combinadas

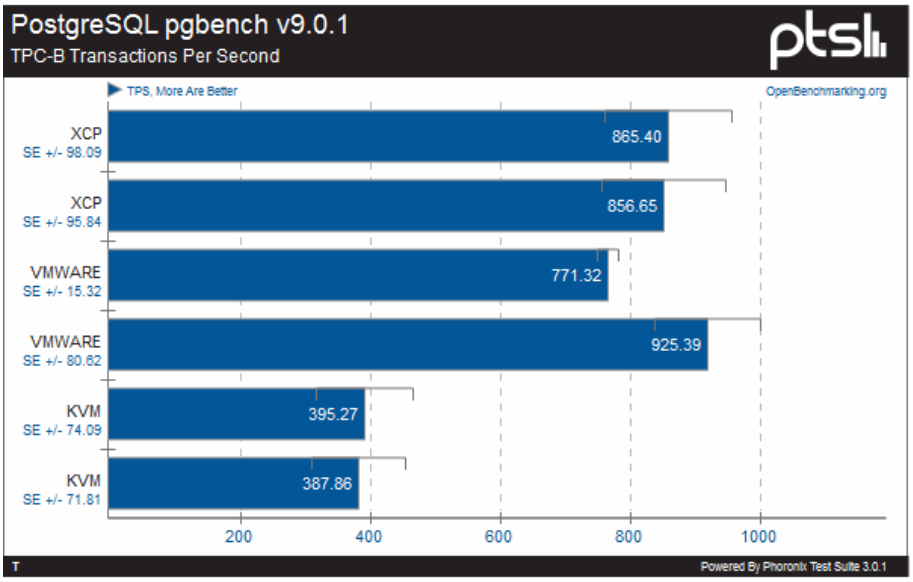
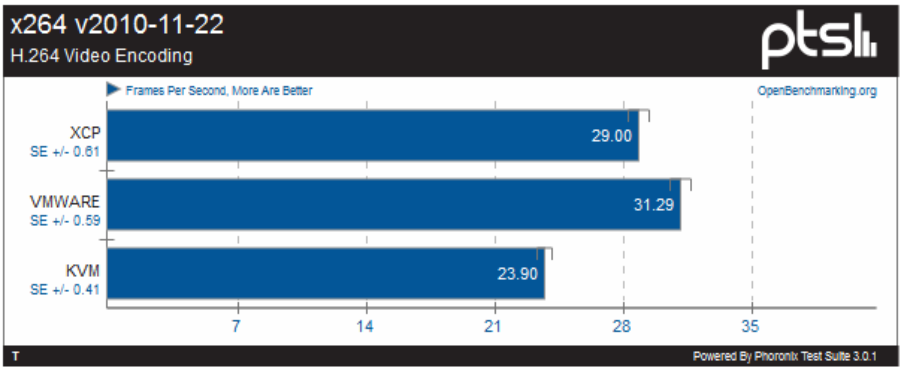


Tabla resultado de las pruebas ejecutando el perfil pts/x264 combinadas



La ejecución de esta prueba supone un giro inesperado en cuanto a los resultados obtenidos. Se puede ver claramente como VMWare, que en las pruebas individuales había puntuado muy bajo en estos perfiles, avanza para convertirse en claro vencedor de esta prueba. El margen de mejora sobre Xen Cloud Platform no es grande, sin embargo si evidente. Esta información clarifica que VMWare está evidentemente optimizado para entornos de alta carga y procesos heterogéneos.

Por otro lado, también sorprende los resultados negativos ofrecidos por KVM sobre ProxMox. Hemos de tener en cuenta que esta plataforma cuenta con una versión muy actualizada de este hipervisor, al contrario que XCP y la versión evaluada de VMWare (que no tienen las últimas versiones de los hipervisores que representan). En conclusión, ProxMox se muestra poco capaz como hipervisor para entornos con una carga variada y elevada. Hemos de recordar que esta solución utiliza el planificador de tareas de Linux para la ejecución de máquinas virtuales y en perspectiva, parece muy incapaz de gestionar varios procesos de tipo guest trabajando con flujos elevados.

1.2.9.3 Conclusiones globales sobre el rendimiento

Tras la ejecución de todas las pruebas, tanto individuales sobre un único servidor virtualizado como la simulación de un entorno heterogéneo de carga han resultado muy satisfactorias. Algunos resultados han sido más sorprendentes que otros, especialmente en la simulación.

Individualmente los hipervisores no ofrecen grandes diferencias. Todos ellos hacen uso de una forma u otra de las extensiones de virtualización creadas por Intel o AMD para facilitar la comunicación de ciertas instrucciones, por lo que en la base, son muy parecidos. Y ello queda reflejado en los resultados.

Las pruebas muestran un gran rendimiento individual por parte de XCP, superando en muchas de ellas a VMWARE. Este último destaca principalmente en los accesos a disco. Y KVM no destaca en ninguna prueba en especial, dejando entender que es un producto aún en fase de maduración (recordemos que tiene poco más de cuatro años de evolución).

Si bien las pruebas individuales no han arrojado mucha información, ocurre lo contrario con la ejecución de la simulación de carga de trabajo. Los resultados obtenidos muestran como KVM no puede dar la talla ente XCP y VMWare. Además, deja patente la gran optimización del producto comercial para este tipo de carga. En cualquier caso, el margen con XCP es muy estrecho y se podría considerar un empate técnico.

Si nos detenemos a considerar el problema de KVM, podría estar en los drivers de acceso a disco, pues de forma general han dado unos resultados muy negativos en las pruebas. O bien, en los problemas que el planificador de procesos de Linux pudiera tener con múltiples procesos con altos requisitos de E/S y CPU (recordemos que el planificador de Linux es el mismo que gestiona procesos y máquinas virtuales en KVM).

1.2.10 Conclusión final

En base al estudio que se ha realizado sobre la instalación del cliente, en la que se ha extraído la información sobre el hardware, software y número de recursos dedicados a la

infraestructura virtual. Y, en conjunto con el análisis de las soluciones de virtualización Open Source, se puede presentar un diseño final basado en las conclusiones obtenidas.

1. Tamaño del entorno. Se trata de un entorno grande, de cientos de máquinas virtuales.
2. Consumo de recursos elevado. La configuración de memoria y CPU de las máquinas alojadas es elevada.
3. Sistemas Operativos homogéneos. El parque de sistemas operativos es muy pequeño. Existiendo únicamente un sabor de Linux y diversas versiones de Windows.
4. Sistemas operativos mayoritariamente Linux. Aproximadamente un 70% de las máquinas virtuales son Linux.

Además, existen algunas características que no se han detectado como críticas en base al tipo de utilización de la plataforma.

1. El uso de la recolocación dinámica de máquinas virtuales en base a la carga o para aliviar el consumo energético es muy poco utilizado.
2. No se hace uso de la tecnología de snapshots.
3. No se realiza reserva de recursos hardware para las máquinas virtuales.

En cambio, otras características si son obligatorias:

1. Posibilidad de crear clústeres para facilitar la gestión.
2. Alta disponibilidad ante caída de alguno de los host físicos que permita el arranque automático de las máquinas virtuales allí alojadas.

De los productos probados a lo largo de la ejecución y redacción de esta memoria, el que puede dar solución a todas las demandas y requisitos impuestos es Xen Cloud Platform.

Las razones principales son:

1. Soporte de todos los sistemas operativos instalados por el cliente.
2. Soporte del sistema operativo Linux paravirtualizado, por lo que mejora sustancialmente el rendimiento.
3. Soporte de clúster con alta disponibilidad (usando scripts de XenServer).
4. Posibilidad de gestión multiplataforma. OpenXenManager, WebOpenXenManager, XenCenter.

Y además, incluye muchas más características que no siendo requisitos, son muy útiles para una gestión de entornos grandes y complicados.

- Backup del entorno.
- Creación de plantillas para despliegue de máquinas virtuales.
- Gestión del mantenimiento de los servidores en clúster.
- Snapshots de las máquinas virtuales en caliente.

Ninguno de los productos probados utilizan las últimas versiones de los hipervisores, pues Xen Cloud Platform utiliza la versión 3.4.2 de Xen (está disponible la 4.1), ProxMox la 0.10 de KVM

(el desarrollo va por la 0.13) y de VMWare se ha utilizado la versión 3.5 de ESX, cuando van por la versión 4. Sin embargo, si destacan por su estabilidad y funcionalidad.

KVM y VMWare siguen un diseño muy similar en cuanto a la virtualización completa. Sin embargo, Xen y VMWare se asemejan en cuanto a la distribución de diversos niveles jerárquicos de seguridad (o dominios en Xen). En estos dos últimos casos, arranca un núcleo o hipervisor que es un pequeño conjunto de instrucciones que se encargan de gestionar los tres sistemas más importantes: memoria, cpu y disco. VMWare arranca una Service Console virtual y Xen ejecuta el anillo 0 o dominio 0.

En líneas generales, son mayores las similitudes que las diferencias. Al final, lo que diferencia un producto de otro son las funcionalidades y servicios que pueden ofrecer sobre sus sistemas de virtualización y que hagan las labores de los técnicos más sencillas y llevaderas.

KVM y en particular ProxMox tienen un futuro prometedor. Si cumplen con su línea de trabajo y son capaces de ofrecer un entorno con alta disponibilidad, serán un producto muy a tener en cuenta.

Sin embargo, hoy en día Xen como hipervisor, y productos como Xen Cloud Platform, parecen estar más alineados con las necesidades de las empresas. Su rendimiento y su capacidad de crecimiento son positivos. Su diseño de Pools y alta disponibilidad posibilita un mayor tiempo operativo de los servicios que se monten sobre ellos. Y la gestión multiplataforma, sencilla y avanzada posibilita que sea utilizado por diversos perfiles de técnicos. Prácticamente no es necesario ejecutar nada vía comandos para disponer de un entorno operativo.

Por tanto, la conclusión que llegamos en esta memoria y que cierra la misma es a favor del software libre, pues ha demostrado ser tanto o más fiable, potente y gestionable que otras soluciones comercial con enormes costes de mantenimiento y soporte. Y en particular con la solución de la comunidad Xen: Xen Cloud Platform.

2 Glosario

Hypervisor o Virtual Machine Monitor (VMM): Es una plataforma de virtualización que permite utilizar, a la vez, múltiples sistemas operativos.

Máquina Virtual o Virtual Machine (VM): Se denominará máquina virtual al servidor completo virtualizado y que se ejecuta sobre un hipervisor de cualquiera de los tipos.

Huésped (Host): El equipo o servidor que ejecuta la parte de software correspondiente al hipervisor y que contiene los dispositivos físicos que se comparten.

Invitado (guest): El software o aplicación que se ejecuta sobre un huésped (host). Puede ser una máquina virtual.

TCO (Total Cost Ownerhip): Coste Total de Propiedad. Es un método de cálculo diseñado para ayudar a los usuarios y a los gestores empresariales a determinar los costes directos e indirectos, así como los beneficios, relacionados con la compra de equipos o programas informáticos.

ROI (Return Of Investment): Retorno de la Inversión. Para luego analizar la incidencia de los excedentes de existencias o de activos obsoletos en él.

HBA: Dentro del hardware, un controlador de host, adaptador de host, o adaptador de bus del host (HBA) conecta un sistema servidor (el Ordenador) a una red y dispositivos de almacenamiento. Normalmente se refieren a dispositivos a los que conectar otro dispositivos IDE, SCSI, Canal de Fibra y eSATA, pero también se suele utilizar el mismo término para los dispositivos que se conectan a sistemas Ethernet, FireWire y USB. Recientemente, la llegada del iSCSI ha dado lugar a HBAs via Ethernet, que se diferencian de las tarjetas de red en que incluyen hardware dedicado para iSCSI.

Fiber Channel: El canal de fibra (del inglés fibre channel) es una tecnología de red utilizada principalmente para redes de almacenamiento, disponible primero a la velocidad de 1 Gbps y posteriormente a 2, 4 y 8 Gbps.

LUN: En almacenamiento, una logical unit number o LUN es una dirección para una unidad de disco duro y por extensión, el disco en sí mismo. El término es originario del protocolo SCSI como una forma de diferenciar unidades de disco individuales dentro de un bus SCSI tal que un array de discos.

Clúster: El término clúster se aplica a los conjuntos o conglomerados de computadoras construidos mediante la utilización de componentes de hardware comunes y que se comportan como si fuesen una única computadora.

HA (High Availability): Es un protocolo de diseño del sistema y su implementación asociada que asegura un cierto grado absoluto de continuidad operacional durante un período de medición dado. Disponibilidad se refiere a la habilidad de la comunidad de usuarios para acceder al sistema, someter nuevos trabajos, actualizar o alterar trabajos existentes o recoger los

resultados de trabajos previos. Si un usuario no puede acceder al sistema se dice que está no disponible. El término tiempo de inactividad (downtime) es usado para definir cuándo el sistema no está disponible.

Virtual Appliance: Es una imagen de una máquina virtual diseñada para ejecutarse sobre una plataforma de virtualización.

NFS: Es un protocolo de nivel de aplicación, según el Modelo OSI. Es utilizado para sistemas de archivos distribuido en un entorno de red de computadoras de área local. Posibilita que distintos sistemas conectados a una misma red accedan a ficheros remotos como si se tratara de locales. Originalmente fue desarrollado en 1984 por Sun Microsystems, con el objetivo de que sea independiente de la máquina, el sistema operativo y el protocolo de transporte, esto fue posible gracias a que está implementado sobre los protocolos XDR (presentación) y ONC RPC (sesión) .1 El protocolo NFS está incluido por defecto en los Sistemas Operativos UNIX y la mayoría de distribuciones Linux.

SnapShots: Es una función de algunos sistemas que realizan copias de seguridad de ficheros almacenándolos tal y como fueron capturados en el pasado.

Bare-meta: Definiremos con este término a una plataforma de virtualización cuyo instalador no requiera ejecutarse sobre un sistema operativo. Permite ser instalado directamente sobre el hardware del servidor.

Multipath: Es una metodología de acceso a disco. En principio surgió como una implementación de acceso SCSI por múltiples vías. Conforma una capa de conmutación para distintos caminos al mismo dispositivo de almacenamiento.

SAN (Storage Area Network): Es una red concebida para conectar servidores, matrices (arrays) de discos y librerías de soporte. Principalmente, está basada en tecnología fibre channel y más recientemente en iSCSI. Su función es la de conectar de manera rápida, segura y fiable los distintos elementos que la conforman.

3 Bibliografía y Referencia

Bibliografía

Hirt, Timo, 2010, KVM - The kernel-based virtual machine. Disponible en: http://www.cs.hs-rm.de/~linn/fachsem0910/hirt/KVM.pdf.

Sabater, Jaume, 2006, Virtualización con Xen en Debian Etch con kernel a medida para 32 y 64 bits. Disponible en: http://www.redes-linux.com/manuales/virtualizacion/xen_debian.pdf.

Armando Medina, Jorge; Sánchez Martínez, Alejandro, 2010, Administración de servidores virtuales con Xen y GNU/Linux. Disponible en: http://tuxjm.net/docs/Administracion_de_Servidores_Virtuales_con_Xen_y_GNU_Linux/html-onechunk/#inatroduccion-a-la-virtualizacion.

Spector, Stephen, 2010, New to Xen Guide. Disponible en: http://www.xen.org/files/Marketing/NewtoXenGuide.pdf.

Arcangeli, Andrea, 2008, Linux as a Hypervisor. Disponible en: http://www.linuxplanet.com/linuxplanet/reports/6503/1/.

Shah, Amit, 2008, DEEP VIRTUE. Disponible en: http://www.linux-magazine.com/Issues/2008/86/DEEP-VIRTUE.

Larabel, Michael; Tippett, Matthew, 2011, Phoronix Test Suite v3.0.0 (Iveland) User Manual. Disponible en: http://www.phoronix-test-suite.com/.

Xen.org Systems, Inc. , 2009, Xen Cloud Platform Administrator's Guide: Release 0.1.

Referencias

http://www.xen.org/products/cloudxen.html

http://blog.xen.org/

http://wiki.xen.org/xenwiki/

http://www.redes-linux.com/manuales/virtualizacion/xen_debian.pdf

http://www.slideshare.net/saghul/virtualizacin-con-xen-y-kvm

http://tuxjm.net/docs/Administracion_de_Servidores_Virtuales_con_Xen_y_GNU_Linux/html-onechunk/#inatroduccion-a-la-virtualizacion

http://www.howtoforge.com/kvm-and-openvz-virtualization-and-cloud-computing-with-proxmox-ve

http://itmanagement.earthweb.com/features/article.php/12297_3779881_1

http://host-sflow.sourceforge.net/

http://blog.sflow.com/2011/01/proxmox.html

http://forum.proxmox.com/threads/1449-Citrix-XENServer-Express-VM-Migration

http://searchsystemschannel.techtarget.com/tutorial/Xen-vs-KVM-and-KVM-migration-guide

http://forum.proxmox.com/archive/index.php/t-401.html?s=d1f36ecefcfbaba1d54b02524ec709d7

http://downloads.vmware.com/d/info/datacenter_downloads/vmware_vmmark/2_0

www.phoronix-test-suite.com

www.openbenchmarking.com

http://wiki.qemu.org/Qemu-doc.html

http://www.worldlingo.com/ma/enwiki/es/OpenVZ

4 Anexos

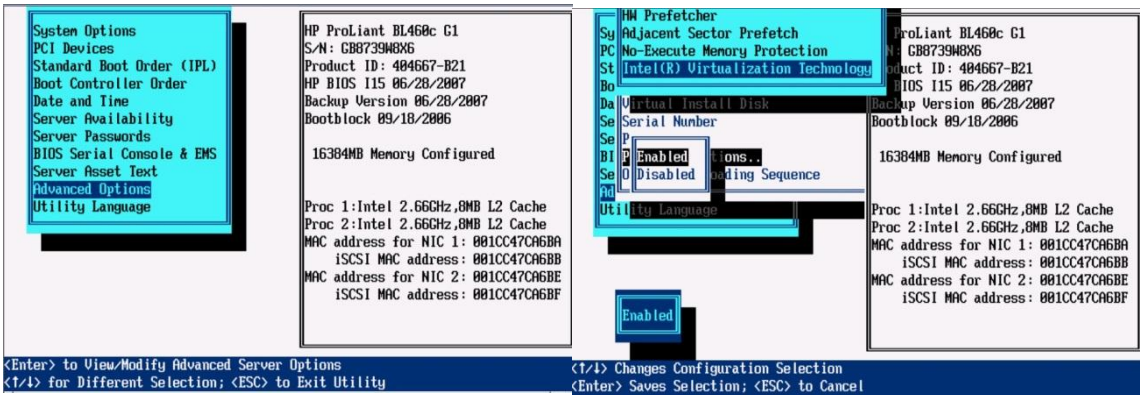
4.1 Preparación de los Servidores Físicos

Antes de realizar cualquier instalación de un hipervisor como XCP, Proxmox o VMWARE en un servidor físico, se ha de confirmar que estos cumplan una serie de requisitos hardware. El mayor de estos es contar con una XPU de 64 bits con soporte al conjunto de instrucciones Intel EM64T o AMD64. En general esto se define como soporte Intel VT o bien AMD-V.

El servidor físico seleccionado (HP BL460c G1) consta de dos CPU Intel X5355 de cuatro Cores cada una. Estas CPU poseen soporte para el conjunto de instrucciones EM64T. Sin embargo, se encuentra desactiva por defecto. Por lo tanto, se ha de habilitar para poder acceder al todas las funciones de virtualización de los productos.

Para activar el soporte Intel VT en estos servidores, se deberá acceder a la BIOS del mismo. Esto se realizará a través de la tarjeta de gestión remota (iLo) y pulsando la combinación de teclas apropiada.

Una vez en la BIOS se accederá a la sección de Advanced Options. Y se activará la opción Intel® Virtualization Technology.

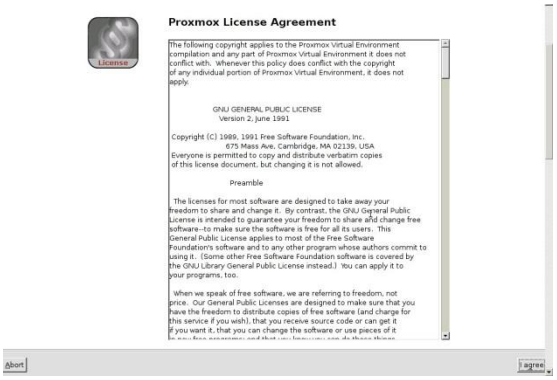


4.2 Instalación ProxMox VE

4.2.1 Instalación de los nodos

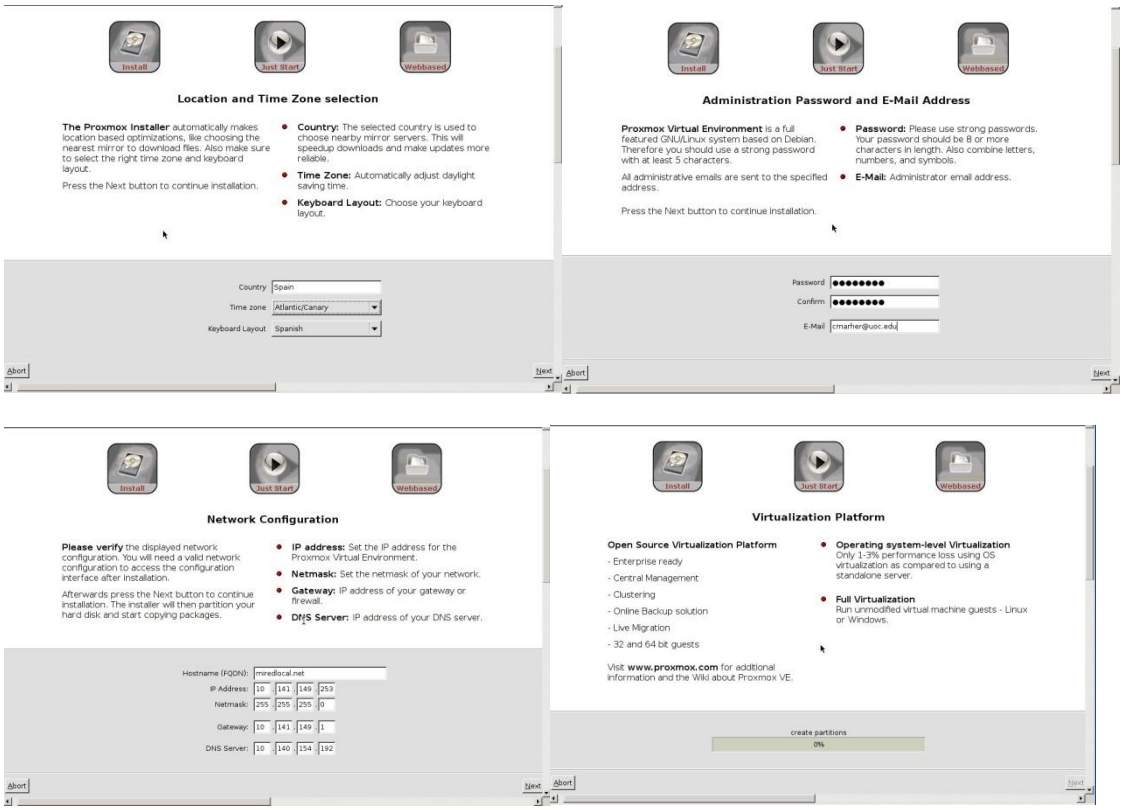
La instalación de la plataforma de virtualización ProxMox en un servidor físico comprende los siguientes pasos.

1. Se ha de descargar la imagen de CD de este sistema. Para lo que se accederá a la página web http://www.proxmox.com y se obtendrá el CD de instalación.
   - En el transcurso de este proyecto, se ha utilizado lo versión **proxmox-ve_1.8-5754-4.iso**.
2. A través de la interfaz iLo (integrated light out) de los servidores físicos, se accede a la consola del servidor y se monta la imagen del CD descargado.
3. Se enciende el servidor y se siguen las instrucciones en pantalla.



- Tras la pantalla de inicio y el acuerdo de licencia, se solicita indicar el disco en el que se desea instalar el producto. En el presente caso, se realiza la instalación en el disco local del servidor.

- A partir de este punto únicamente se solicitará los parámetros de configuración concernientes a la configuración horaria, password del administrador y configuración de red del nuevo nodo.



- Finalmente, la instalación concluye sin mayores complicaciones. Se reinicia el servidor y ya dispondríamos de una plataforma de virtualización operativa.

4.2.2 Configuración del clúster

Para montar un clúster con esta solución, se deberá instalar otro nodo con diferente nombre y configuración de red. Para lo que se seguirá el proceso descrito con anterioridad. Una vez instalado y hayamos arrancado el nuevo servidor.

Se ha de tener en cuenta que proxmox utiliza una arquitectura de clúster con un único servidor maestro (master) que es quien controla todos los recursos y la disponibilidad del sistema. En futuras versiones se espera que esta arquitectura evolucione a un entorno multimaster sin las limitaciones del actual sistema.

Por lo tanto, designaremos una máquina que sea el maestro del clúster y desde ella se ejecutarán los comandos de creación del clúster.

La creación del clúster no está implementada a través de la interfaz gráfica, por lo que se accederá vía consola de comandos (ssh) al servidor seleccionado.

Un requisito indispensable para la formación del servicio de clúster es la sincronización horaria de todos los servidores que lo conformarán. Por defecto, cada uno de los nodos posee un cliente/servidor NTP configurado. Deberemos sincronizarlos previa a la creación del clúster. En

caso de estar sincronizados, el clúster no se creará correctamente. Sin embargo, una vez sincronizados este comenzará a funcionar.

El clúster se formará mediante el comando *pveca* de la siguiente manera:

- Ejecutamos el siguiente comando, lo cual designará al maestro.

```
$pveca -c
```

- Se verifica el estado del clúster (por ahora con un único nodo)

```
$pveca –l
```

- Se añade un nuevo nodo al clúster.

```
$pveca -a -h <IP-DEL-NUEVO-NODO>
```

Previo a este paso, no es una mala política añadir los nombres de todos los nodos que se desean añadir al clúster en el fichero /etc/hosts de la máquina. En general, en todos los nodos que involucran al clúster.

- Se podrá verificar nuevamente el estado del clúster con el comando señalado anteriormente.

```
$pveca -l
```

4.2.3 Configuración del disco de la SAN

Actualizamos la instalación de proxmox realizada. Se ejecutan los siguientes comandos en todos los componentes del clúster.

```
$aptitude update
```

Se instala el módulo de multipath para la gestión de la LUN que se ha añadido vía las HBA del equipo.

```
$aptitude install multipath-tools
```

Desde cualquiera de los nodos, se ejecutará el comando multipath para identificar el disco que se ha asignado a los servidores.

```
proxmox1:/mnt/pve# multipath -ll
360050768019081acf80000000000022bdm-24 IBM     ,2145
[size=150G][features=1 queue_if_no_path][hwhandler=0]
\_ round-robin 0 [prio=100][active]
 \_ 0:0:4:0 sda 8:0   [active][ready]
 \_ 1:0:4:0 sde 8:64  [active][ready]
\_ round-robin 0 [prio=20][enabled]
 \_ 0:0:5:0 sdc 8:32  [active][ready]
 \_ 1:0:5:0 sdg 8:96  [active][ready]
360050768019081acf80000000000230dm-25 IBM     ,2145
[size=150G][features=1 queue_if_no_path][hwhandler=0]
\_ round-robin 0 [prio=100][active]
 \_ 0:0:5:1 sdd 8:48  [active][ready]
```

```
 \_ 1:0:5:1 sdh 8:112 [active][ready]
\_ round-robin 0 [prio=20][enabled]
 \_ 0:0:4:1 sdb 8:16  [active][ready]
 \_ 1:0:4:1 sdf 8:80  [active][ready]
```

Se observa la existencia de dos LUN. Cada una de ellas, es visible por cuatro canales. Esto es debido a que el servidor dispone de dos tarjetas HBA con dos puertos cada una.

La consecuencia de cara al sistema, es la generación de un dispositivo en /dev/mapper/360050768019081acf800000000000230 que enmascara los múltiples caminos que existen para llegar al disco.

En este punto, y conocido el nombre lógico del dispositivo que se quiere utilizar, se creará un volumen LVM que será el utilizado para almacenar las máquinas virtuales.

Se creará un nuevo volumen LVM mediante los siguientes pasos:

- Creamos un disco físico asociado al nombre lógico del disco de la SAN.

```
$pvcreate /dev/mapper/360050768019081acf800000000000230
```

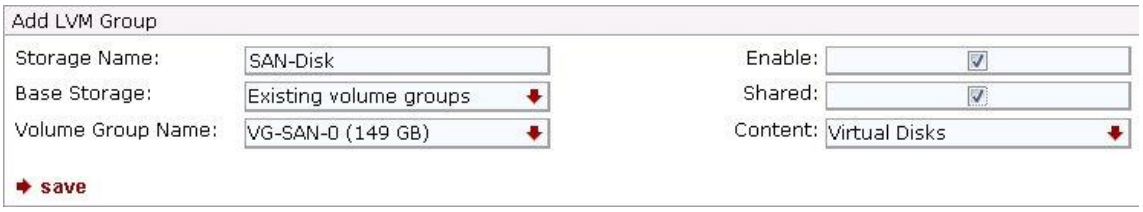
- Creamos un Volume Group que contenga este nuevo disco creado.

```
$vgcreate VG-SAN-0 /dev/mapper/360050768019081acf800000000000230
```

Tras este paso, se ha generado el Volume Group VG-SAN-0 y lo podremos añadir al servicio proxmox de nuestra plataforma. Se realizará desde uno de los nodos a través de la interfaz gráfica.

- Se selecciona la opción *Storage* de la sección de configuración de la interfaz.

- Desplegamos las opciones y añadimos un nuevo LVM Group.



- Se introduce el nombre del nuevo grupo "SAN-Disk" y se observará como el nuevo volumen que hemos creado nos aparece automáticamente en el desplegable. Dado que la instalación es en clúster, marcaremos la opción *Shared*. El tipo de contenido de este disco se designará como "Virtual Disks", pues lo que se desea es instalar máquinas virtuales.
- Finalmente, al grabar los datos anteriores, tendremos un nuevo volumen donde se podrán crear nuevas máquinas virtuales.

Storage List

Storage Name	Type	Enabled	Active	Shared	Used (GB)	Capacity (GB)	
local	Directory	Yes	Yes	No	0.47	37.73	1.25%
CentOS-ISO	NFS	Yes	Yes	Yes	280.84	480.00	58.51%
Backup	Directory	Yes	Yes	No	280.84	480.00	58.51%
SAN-Disk	LVM	Yes	Yes	Yes	8.00	150.00	5.34%

4.2.4 Repositorio de ISO de Sistemas Operativos

El siguiente paso para generar un entorno virtual operativo es disponer de un repositorio de imágenes ISO necesarias para poder instalar sistemas operativos en las máquinas virtuales.

Proxmox y su flexibilidad de modelos de almacenamiento, permite realizar esta tarea de múltiples formas. Además, el repositorio de imágenes puede estar compartido en diversos soportes. A través de la interfaz web, se podrán seleccionar dos tipos diferentes, o bien NFS o bien mediante un directorio existente en el sistema operativo anfitrión. Este último directorio puede ser local, o remoto. En caso de ser remoto, se puede utilizar cualquier protocolo soportado por la distribución Debian Lenny sobre la que está basada la plataforma (NFS, CIFS, etc).

Se ha optado por una configuración vía NFS y se ha realizado directamente a través de la interfaz web. Para ello se ejecutarán los siguientes pasos:

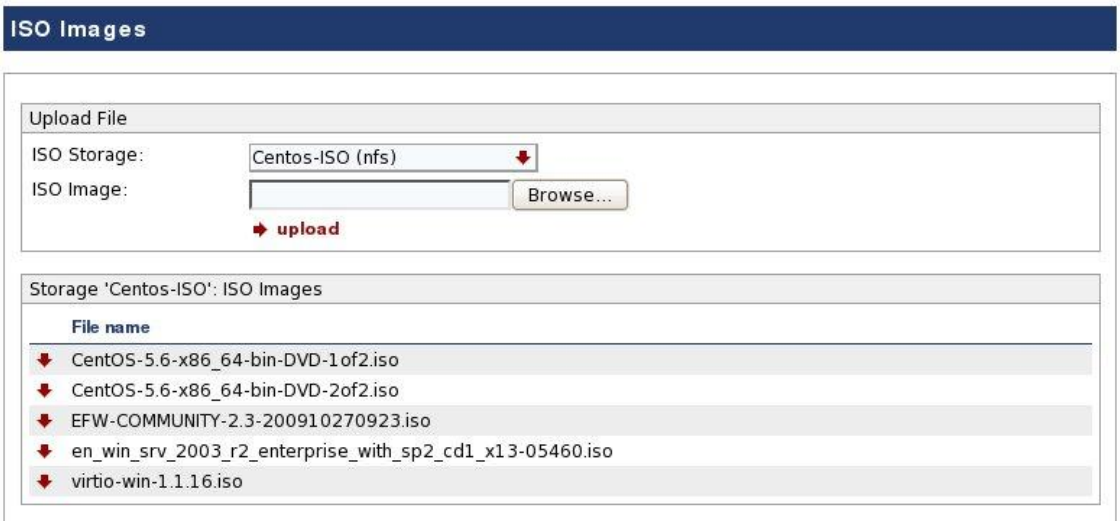
Desde la pantalla principal de gestión se selecciona la opción Storage. Esta opción nos abrirá el formulario principal de gestión de los sistemas de almacenamiento soportado y montados.

Se añadirá un nuevo recurso compartido de tipo NFS a través de la opción disponible en el menú desplegable que surge al pulsar sobre la opción Storage List.

Se abrirá una nueva pantalla de configuración que permite la creación del nuevo recurso tras completar el formulario con la información solicitada.

En este formulario se indicará el nombre con el que se hará referencia al recurso compartido, el tipo de contenido que tendrá el almacenamiento (imágenes ISO) y la información relativa al servidor NFS que se configurará (IP y ruta al directorio).

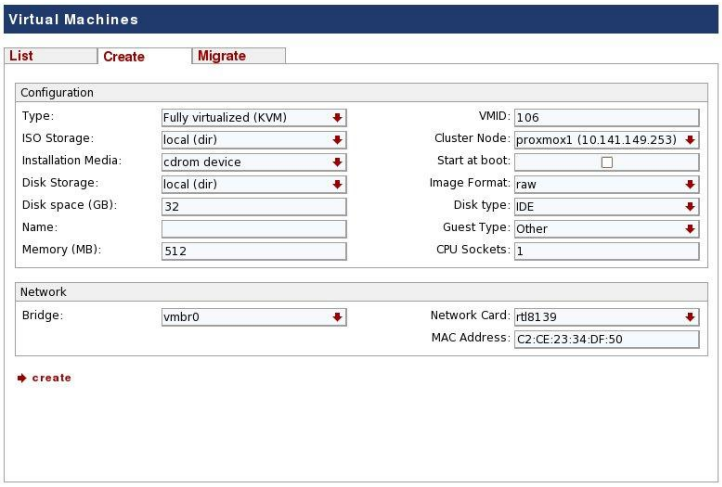
Completada esta información, se dispondrá de un contenedor de imágenes listo para ser utilizadas por parte de las máquinas virtuales.



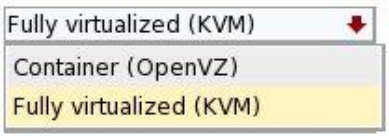
4.2.5 Creación de Máquinas Virtuales

Llegados a este punto se dispone de todo lo necesario para comenzar la creación de nuevas máquinas virtuales en el sistema Proxmox VE. Esta tarea se puede realizar de forma rápida y sencilla desde la interfaz web del producto. Un ejemplo de creación de una máquina virtual como las utilizadas para las pruebas en este proyecto es la siguiente.

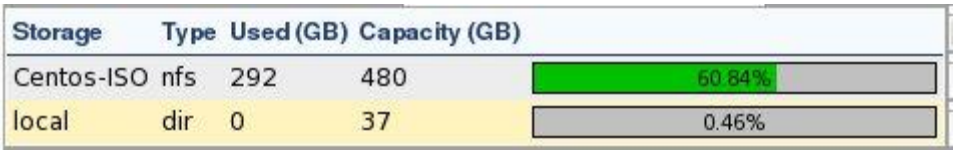
Unja vez logados en la interfaz de gestión se seleccionará la opción *Virtual Machines* del menú de la izquierda, en la sección VM Manager. Se nos mostrará una nueva pantalla con todas las máquinas virtuales existentes en el sistema se tendrá una pestaña dedicada a la creación de nuevas máquinas. Una se ha pulsado sobre esta pestaña (*Create*) se deberá rellenar toda la información necesaria.



Type: ProxMox nos permite seleccionar dos tipos distintos de máquinas virtuales. Virtualización completa con KVM o mediante el contenedor de virtualización de sistema operativo OpenVZ.

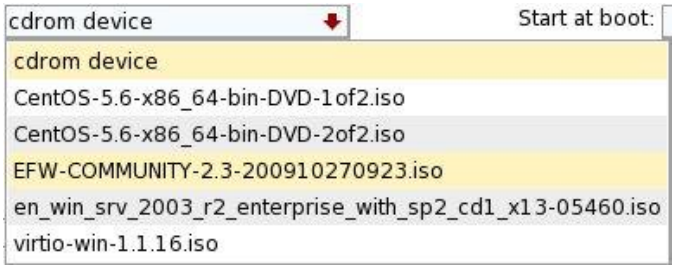


ISO Storage: para instalar el sistema operativo en la nueva máquina virtual, se puede seleccionar alguno de los repositorios de imágenes ISO que permitirá seleccionar la misma desde la siguiente opción: Installation Media.

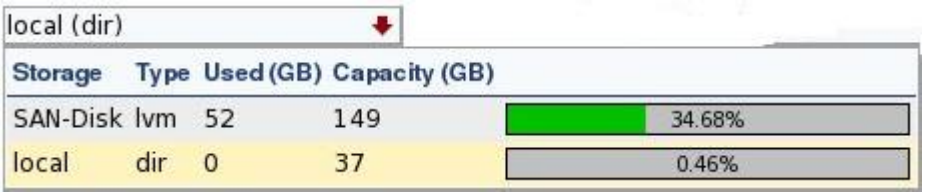


Storage	Type	Used (GB)	Capacity (GB)	
Centos-ISO	nfs	292	480	60.84%
local	dir	0	37	0.46%

Installation Media: una vez seleccionado el repositorio, se mostrarán las ISO que existan en el mismo y se podrá seleccionar aquella que se requiera para la instalación.



Disk Storage: se mostrarán todos los dispositivos de almacenamiento disponibles en el sistema y de ellos se seleccionará aquel que se quiere que contenga la máquina virtual.



Disk Space: se introducirá un valor en Gb que expresará el tamaño del disco virtual que se asignará a la nueva máquina.

Name: El nombre lógico que tendrá la máquina virtual.

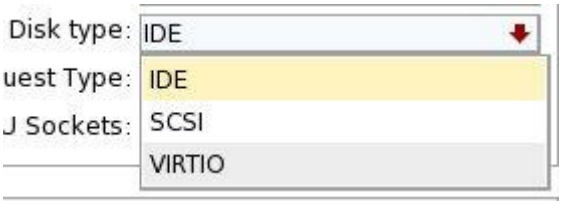
Memory: la cantidad de memoria que se le asignará a la nueva máquina virtual. Valor expresado en Mb.

VMID: el identificador único de la máquina virtual. El sistema asignará automáticamente un identificador no usado, pero se podrá configurar uno nuevo.

Clúster Node: se indicará a que nodo del clúster pertenece la máquina virtual.

Start at boot: se puede marcar esta opción para que la máquina virtual arranque cuando reiniciemos el servidor anfitrión.

Image format: proxmox permite seleccionar el formato de la imagen de disco que se almacenará. Concretamente qcow2 es el formato nativo de qemu y soporta un gran número de características avanzadas (snapshots, encriptación, compresión). Es un formato ideal cuando utilizamos almacenamiento remoto del tipo NFS o CIFS. Vmdk es el formato de Vmware y permite migraciones de máquinas entre estos productos. Finalmente, raw es el formato por defecto y la información se almacena en formato binario plano.



Disk type: se puede seleccionar el tipo de emulación del hardware de disco. IDE es la más compatible y VIRTIO es la que tiene mayor rendimiento, pero requiere instalar los drivers paravirtualizados en el sistema operativo. Estos drivers, pueden o no estar disponible para todos los sistemas soportados.

Guest Type: para poder optimizar el rendimiento de la máquina virtual, se ha de seleccionar el tipo de sistema operativo a instalar en la misma. Puede ser: Linux 2.4, Linux 2.6, Other, Windows 2000, Windows 2003, Windows 2008, Windows Vista, Windows XP.

CPU Sockets: número de procesadores virtuales que se asignará a la máquina virtual

Bridge: dispositivo de red que se asociará a la interfaz de red virtual de la máquina virtual. Se desplegará un listado con todos los dispositivos disponibles.

Network Card: se seleccionará el tipo de emulación del driver de la tarjeta de red virtual de la máquina virtual. Se despliegan todas las disponibles. Para mayor rendimiento se puede optar por el formato VIRTIO, pero al igual que con el disco, requiere de la instalación de los pertinentes drivers paravirtualizados en el sistema operativo guest.

MAC Address: el asistente asigna de forma automática una MAC aleatoria para la nueva interfaz de red virtual. Si se desea, se puede asignar una nueva manualmente.

Una vez se ha rellenado completamente el formulario de creación de la máquina virtual, se pulsará sobre la opción *créate* y el sistema generará una nueva máquina virtual con las características detalladas. La nueva máquina aparecerá listada en la sección VM Manager.

4.2.6 Clonado de Servidores

El clonado de servidores es una utilidad que debería incluir cualquier software de virtualización. A través de la plataforma Proxmox VE es posible, pero no lo suficientemente práctico para tener una gran utilidad.

Esta acción es muy útil para el desarrollo de esta memoria, pues resulta sencillo desplegar múltiples máquinas similares si está disponible.

La única forma de realizarla mediante Proxmox es ejecutando una copia de seguridad de la máquina virtual y recuperarla posteriormente con otro nombre (o VMID). Para ello, necesitamos crear un nuevo almacenamiento de tipo backup en el sistema y posteriormente programar una copia de seguridad de la máquina que se desea clonar.

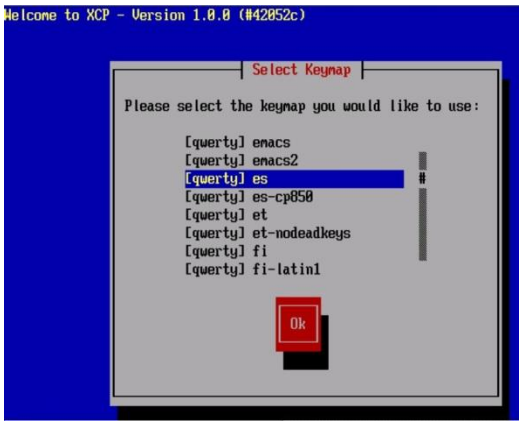
Este método funciona, pero es costoso tanto en espacio como en tiempo. Se ha de almacenar la copia de seguridad y tanto la realización de la misma como la recuperación pueden llegar a tardar mucho tiempo dependiendo del tamaño de los discos virtuales.

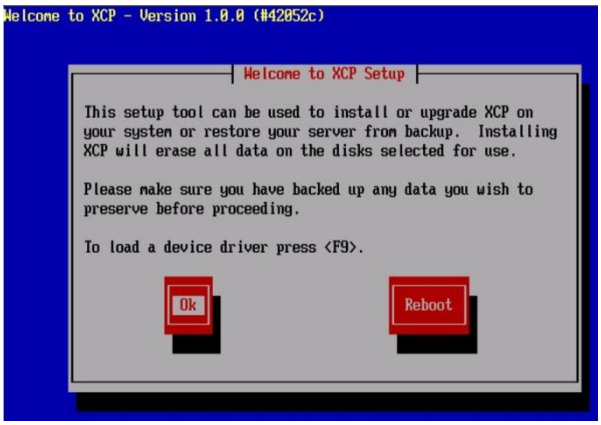
4.3 Instalación de Xen Cloud Platform (XCP)

4.3.1 Instalación de los nodos

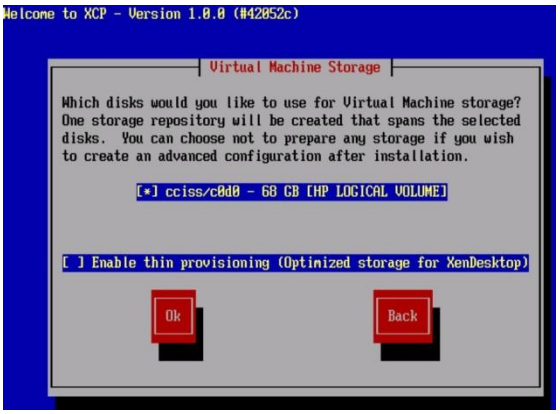
La instalación de la plataforma de virtualización Xen Cloud Platform (XCP) en un servidor físico comprende los siguientes pasos.

1. Se ha de descargar la imagen de CD de este sistema. Para lo que se accederá a la página web http;//www.xen.org y se obtendrá el CD de instalación.
   - En el transcurso de este proyecto, se ha utilizado lo versión **XCP-1.0-base-42052.iso**.
2. A través de la interfaz iLo (integrated light out) de los servidores físicos, se accede a la consola del servidor y se monta la imagen del CD descargado.
3. Se enciende el servidor y se siguen las instrucciones en pantalla.

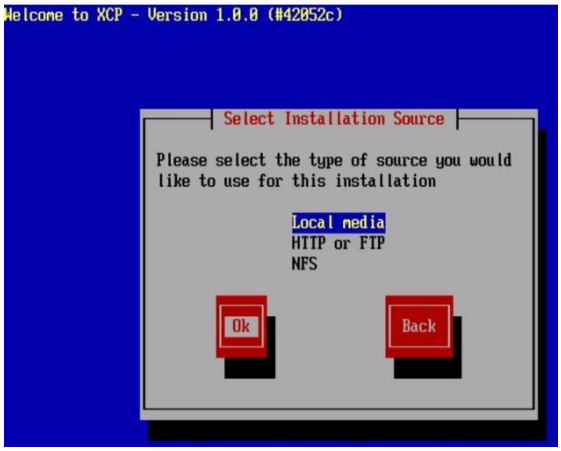


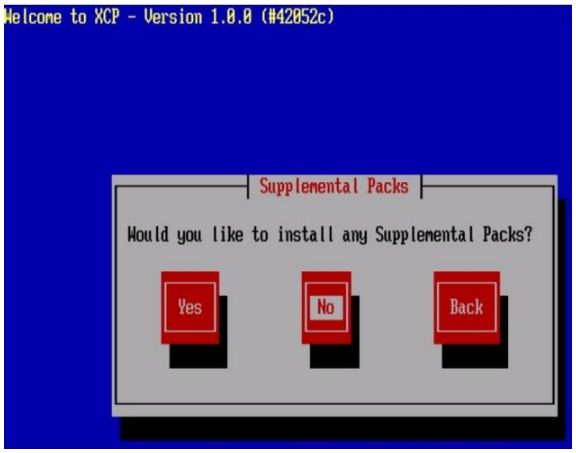


- Se selecciona el idioma y formato del teclado y aceptamos el aviso que indica la eliminación de toda información de los disco que se seleccionen durante la instalación.

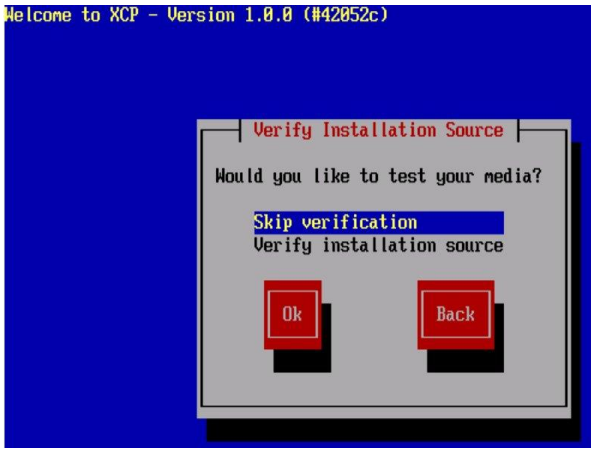


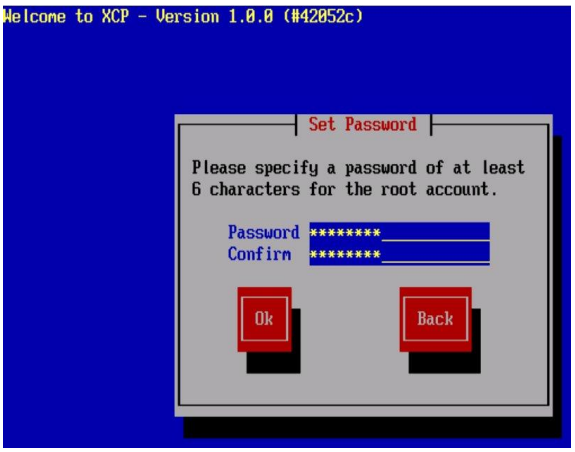
- Se acepta el acuerdo de licencia que afecta a los componentes que conforman el producto. Y posteriormente, seleccionamos el disco local en el que se desea realizar la instalación del nodo XCP.



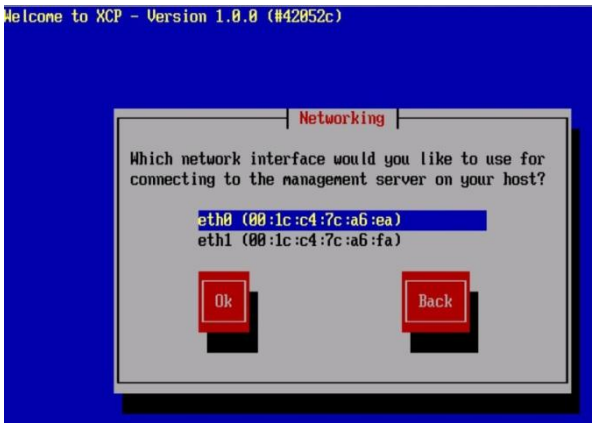


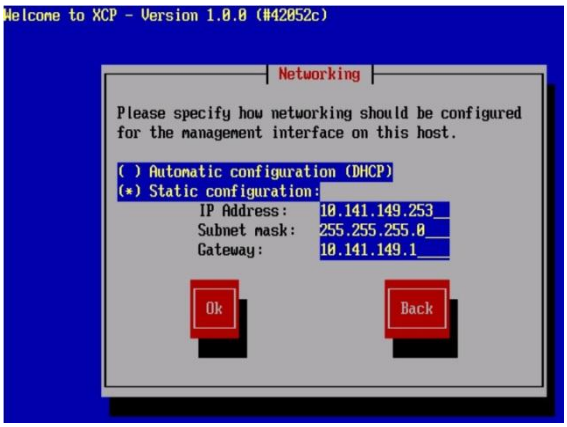
- Se indica el origen de la instalación. En este caso, optamos por una instalación desde el CD local (montado a través de la iLo). En caso de instalaciones grandes, s podría realizar una instalación desatendida a través de NFS o HTTP/FTP. A continuación, se indica que no se instale ningún paquete de suplementos. Esta opción (heredada de XenServer) permitiría añadir nuevas funcionalidades o plantillas de sistemas operativos.

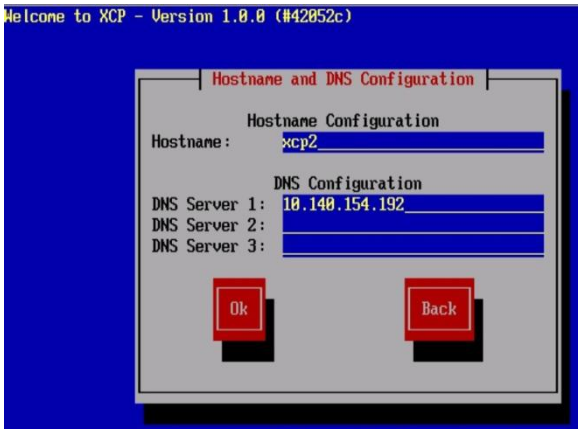




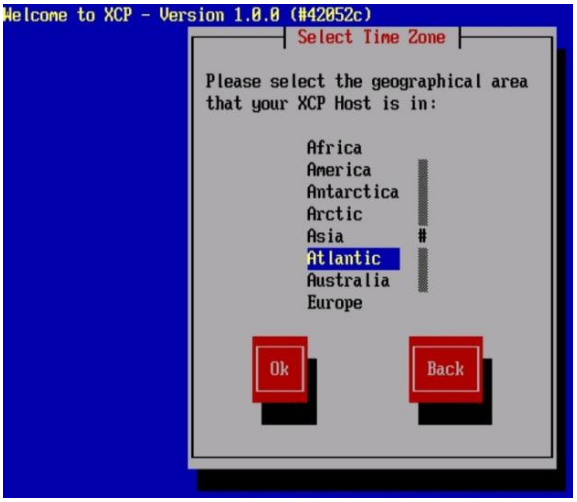
- Al haberse seleccionado una instalación a través del CD local, se nos solicita realizar una verificación del medio. En este caso, se pasará de esta verificación. Además, se nos solicita una contraseña que definirá al usuario administrador del nodo que se está instalando.

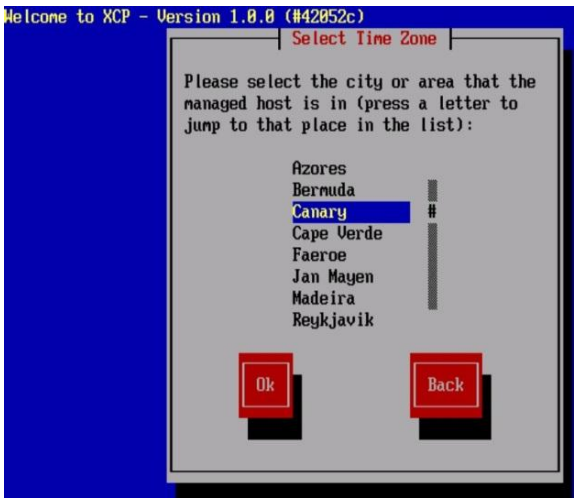


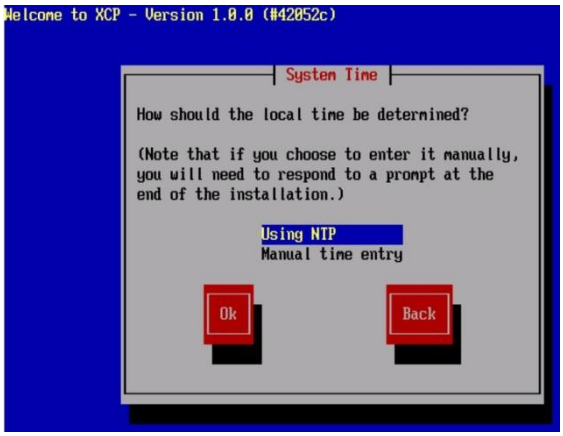


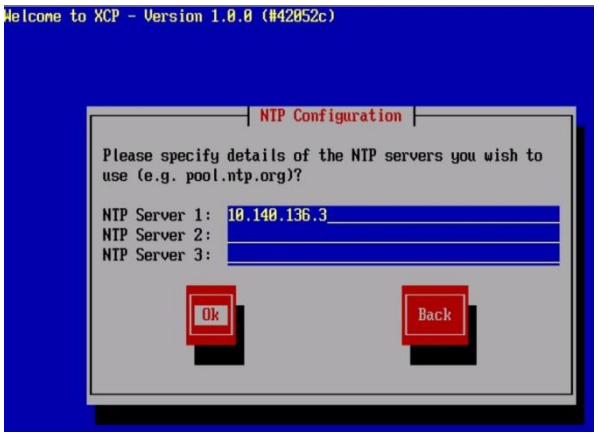


- El siguiente paso de la instalación de XCP está dedicado a la configuración de la interfaz de red de gestión. Se nos solicitará indicar qué interfaz se desea activar para poder gestionar el nodo. Una vez seleccionada, configuraremos dicha interfaz de forma estática o dinámica. Finalmente, indicaremos el nombre del nodo y la dirección IP de un servidor DNS.

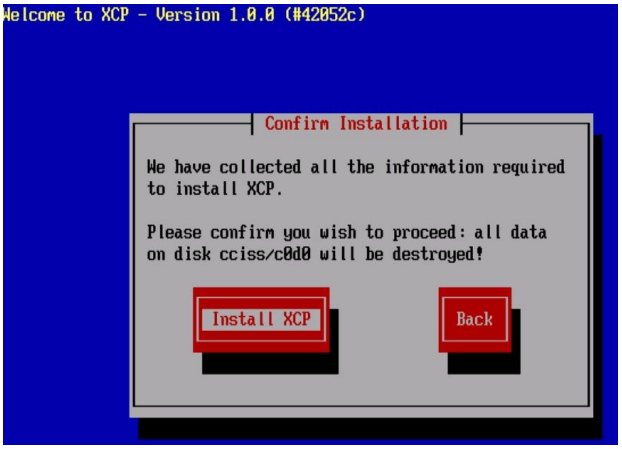


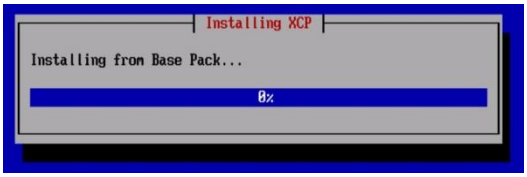




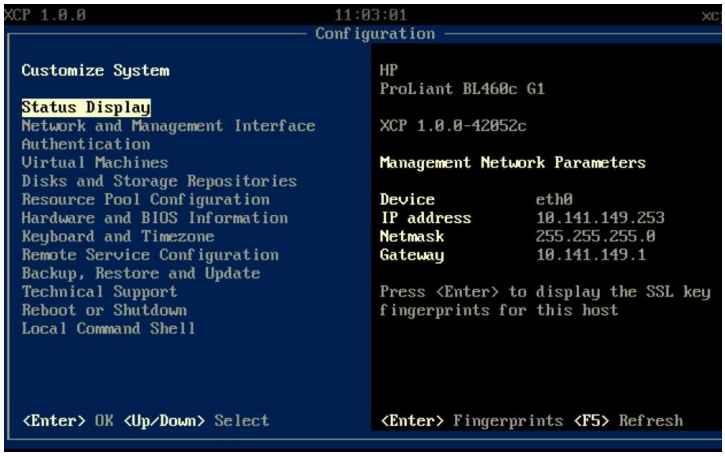


- El último paso de la instalación se trata de definir la configuración horaria del nodo. Se establecerá la zona horaria y en caso de contar con uno, se añadirá un servidor NTP contra el que se sincronizará el servicio.





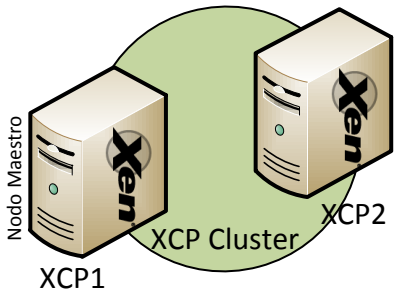
- Una vez finalizada la instalación del servidor, este se reinicia y nos aparecerá la pantalla de gestión. A través de la cual, se tendrá acceso a información de configuración del nodo. Además, se podrán realizar diversas acciones sobre los servicios y máquinas virtuales instaladas. Así, como tener acceso a la consola del dom0.



4.3.2 Creación del Clúster (Pool)

En la terminología de Xen Cloud Platform, y en Xen en general, se designa la estructura de clúster como Pool de recursos. Para la creación del mismo, se podrá proceder de diversas formas, o bien asistidos mediante una GUI, o bien a través del CLI. En este caso, se optará por crear el clúster a través del CLI, y ser agnósticos a cualquier GUI (al menos en este apartado).

Los Pool de recursos en Xen Cloud Platform no son multimaster. Se ha de designar una máquina que asuma el rol de master del Pool. Se generará la siguiente estructura, un Pool que se denominará "XCP Clúster" que contendrá dos nodos XCP que se llamarán XCP1 y XCP2.



Para crear este Pool ejecutaremos una serie de comandos desde el CLI de los nodos que lo compondrán.

- Desde el nodo secundario (XCP2), aquel que no se convertirá en el master del pool, se ejecuta el siguiente comando:

```
$xe pool-join master-address=xcp1 master-username=root master-
password=******
```

- Por defecto, el Pool no tienen nombre. Se deberá asignar un nombre al mismo. Para ello se ejecuta (desde cualquier nodo):

```
$xe pool-param-set name-label="XCP Clúster" uuid=<se autocompletará
con TAB>
```

- Se verificará la creación del Pool mediante el comando:

```
$xe pool-list
```

Se puede observar lo sencillo que supone crear un Pool (clúster) de servidores XCP. Gracias a este Pool, se podrá centralizar la administración de un conjunto de máquinas y sus recursos: disco, imágenes de sistemas operativos, configuración de red, etc.

4.3.3 Creación del almacenamiento compartido para máquinas virtuales

Una vez creado el Pool (XCP Clúster) y si se desean crear máquinas virtuales que puedan migrarse entre los nodos, se deberá crear una unidad de almacenamiento que sea compartida

por todos los nodos. XCP soporta tres tipos distintos de almacenamiento compartido: NFS, hardware HBA o iSCSI.

La solución sobre NFS puede ser de gran utilidad para entornos que no requieren gran rendimiento de disco y cuando no se dispone de tecnologías más avanzadas y costosas. Además, las máquinas virtuales almacenadas mediante este protocolo utilizan el modelo de disco virtual VHD que optimiza enormemente su velocidad sobre protocolos distribuidos en red.

Sin embargo, para el presente proyecto se dispone de disco compartido a través de las interfaces HBA de Fiber Channel de los servidores. Lo cual, permite acceder de forma compartida al almacenamiento a través de una red de fibra con gran velocidad de acceso.

Para configurar este disco, y desde cualquiera de los nodos del Pool, procederemos con los siguientes comandos:

- Se realiza un barrido de las LUN que estén publicadas a través de la ZONA y que sean visibles por las HBA.

```
$xe sr-probe type=lvmohba
<Devlist>
       <BlockDevice>
              <path>
                     /dev/sdd [sdf] [sdb] [sdh]
              </path>
              <numpaths>
                     4
              </numpaths>
              <SCSIid>
                     360050768019081acf800000000000230
              </SCSIid>
              <vendor>
                     IBM
              </vendor>
              <serial>
                     020064206b3eXX03
              </serial>
              <size>
                     161061273600
              </size>
              <adapter>
                     0
              </adapter>
              <channel>
                     0
              </channel>
              <id>
                     5
              </id>
              <lun>
                     1
              </lun>
              <hba>
                     qlogic
              </hba>
       </BlockDevice>
       <BlockDevice>
              <path>
                     /dev/sde [sdg] [sda] [sdc]
              </path>
              <numpaths>
                     4
              </numpaths>
              <SCSIid>
                     360050768019081acf80000000000022b
              </SCSIid>
```

```
            <vendor>
                    IBM
            </vendor>
            <serial>
                    020064206b3eXX03
            </serial>
            <size>
                    161061273600
            </size>
            <adapter>
                    1
            </adapter>
            <channel>
                    0
            </channel>
            <id>
                    4
            </id>
            <lun>
                    0
            </lun>
            <hba>
                    qlogic
            </hba>
    </BlockDevice>
    <Adapter>
            <host>
                    host1
            </host>
            <name>
                    qlogic
            </name>
            <manufacturer>
                    QLogic HBA Driver
            </manufacturer>
            <id>
                    1
            </id>
    </Adapter>
    <Adapter>
            <host>
                    host0
            </host>
            <name>
                    qlogic
            </name>
            <manufacturer>
                    QLogic HBA Driver
            </manufacturer>
            <id>
                    0
            </id>
    </Adapter>
</Devlist>
```

- Se observa la existencia de dos LUN. Una con identificador 360050768019081acf80000000000022b y la otra con 360050768019081acf800000000000230. Cada una de ellas visibles por cuatro caminos diferentes.
- Antes de crear el recurso compartido, se ha de activar el multipathing en los dos nodos.

```
$xe host-param-set other-config:multipathhandle=dmp \
uuid=8624fb61-db3a-48ff-abc8-23949c0c3117
$xe host-param-set other-config:multipathing=true \
uuid=8624fb61-db3a-48ff-abc8-23949c0c3117
$xe host-param-set other-config:multipathhandle=dmp \
uuid=103a3544-ecda-4815-acc7-986066fb60ca
```

```
$xe host-param-set other-config:multipathing=true \
uuid=103a3544-ecda-4815-acc7-986066fb60ca
```

- Creamos el almacenamiento compartido.

```
xe sr-create host-uuid=103a3544-ecda-4815-acc7-986066fb60ca \
content-type=user \
name-label="Shared SAN-Disk" shared=true \
device-config:SCSIid=360050768019081acf80000000000022b type=lvmohba
```

Nos devuelve el uuid del Storage Repository creado:

```
63580d12-a467-5419-f3ed-9724a8d5b0e7
```

- Se listan los Repositorios de Almacenamiento existentes para verificar la existencia del recién creado.

```
# xe sr-list uuid=63580d12-a467-5419-f3ed-9724a8d5b0e7
uuid ( RO)                : 63580d12-a467-5419-f3ed-9724a8d5b0e7
          name-label ( RW): Shared SAN-Disk
    name-description ( RW):
                host ( RO): xcp1
                type ( RO): lvmohba
        content-type ( RO): user
```

4.3.4 Habilitar el servicio de Alta Disponibilidad

El servicio de Alta Disponibilidad permite a la plataforma de virtualización controlar el arranque de los servidores virtuales ante la caída de alguno de los nodos huésped. Para poder configurarlo en XCP es necesario contar con almacenamiento compartido, bien a través de iSCSI o bien mediante disco ofrecido por HBA. Además, las máquinas virtuales que se encuentran protegidas, se han de crear en un disco compartido por todos los nodos del Pool.

Por defecto, la plataforma XCP no contempla el servicio de Alta Disponibilidad, aunque la pila de gestión xapi contiene todo lo necesario para su funcionamiento. Este servicio está disponible en el producto comercial sobre el que está basada la plataforma open-source XCP, la solución de Citrix XenServer.

Durante la realización de esta maqueta, y tras realizar algunas pruebas, se ha logrado dotar a la plataforma XCP de esta funcionalidad. Para ello, se ha copiado el directorio en el que se encuentran los scripts de control del servicio de Alta Disponibilidad de un entorno Citrix XenServer (gratuito, pero con este servicio deshabilitado) en la misma ubicación en el entorno open-source Xen Cloud Platform (/opt/xensource/xha).

Una vez copiados y tras asignarles permisos de ejecución, se ha logrado configurar sin problemas el servicio de Alta Disponibilidad. Este proceso tiene como requisito tener disponible un disco (Fiber Channel HBA o iSCSI) compartido entre todos los integrantes del Pool. En este caso, haremos uso del disco previamente configurado.

Para activar la Alta Disponibilidad se ejecutarán los siguientes comandos desde cualquiera de los nodos:

4.3.5 Entornos de Gestión

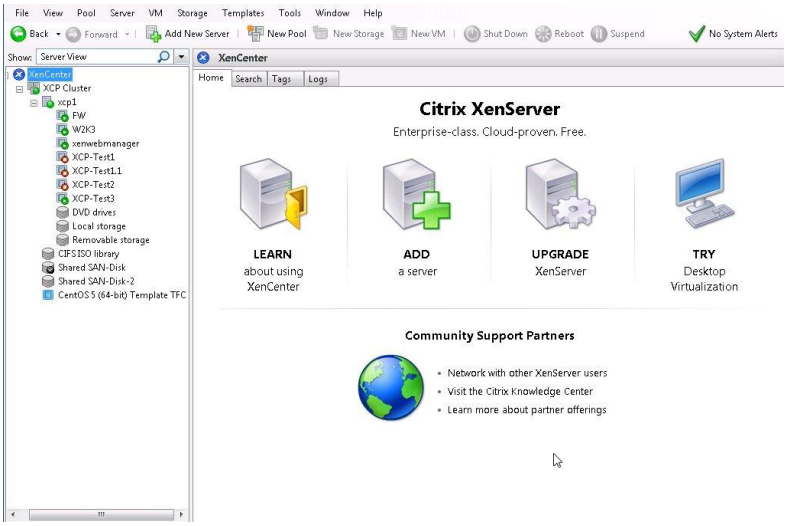
Una vez finalizada la instalación de XCP, se ha de optar por alguna forma de gestión de la plataforma. Aunque por defecto tenemos la interfaz de comandos "xe" disponible en todos los nodos del Pools, se muestra ineficaz de cara a una gestión de gran cantidad de nodos y máquinas virtuales.

Por lo tanto, es una tarea importante decidir que herramientas se utilizarán para aliviar las tareas normales de gestión sobre el entorno virtual.

De partida, dentro del proyecto XCP de la comunidad Xen, se pueden escoger diversas alternativas.

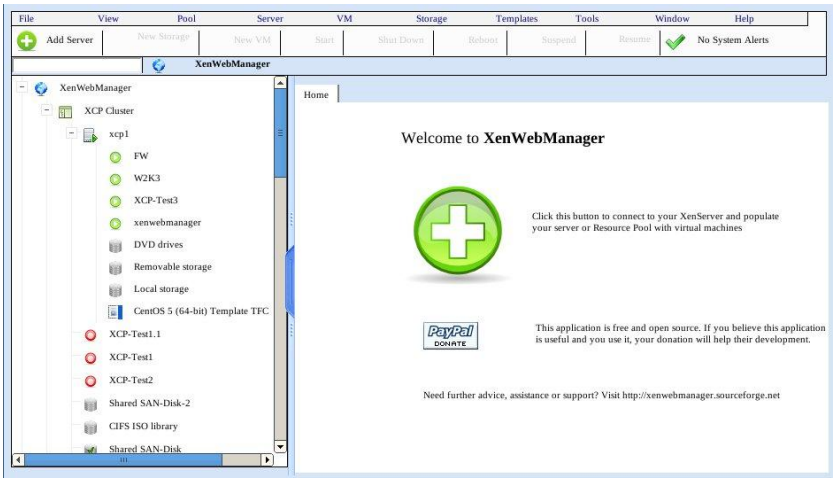
- OpenXenManager.
- WebOpenXenmanager.
- Citrix XenCenter.

Para el desarrollo de este proyecto se ha utilizado la herramienta comercial, pero gratuita XenCenter. Disponible para su descarga en la web de esta compañía. El mayor inconveniente que posee es su dependencia de un entorno Windows sobre el que se ha de instalar. Sin embargo, es la que nos aporta la mayor capacidad de gestión sobre todo el entorno.



Basados en esta misma herramienta, pero con una orientación multiplataforma, son las dos siguientes herramientas. La primera, OpenXenManager consiste en una serie de ejecutables desarrollados en python. Y por tanto, ejecutables desde cualquier sistemas con una base python instalada. La segunda, WebOpenXenManager es un desarrollo de la primera preparado para montarse sobre un servidor web. Una de las ventajas de este último es la disponibilidad de un Appliance Virtual ya configurado y listo para ser utilizado.

Si no es posible utilizar la herramienta XenCenter de Citrix, la aproximación recomendada sería un despliegue del Appliance WebOpenXenManager. Una vez desplegado el mismo y configurado el direccionamiento de la máquina virtual, podremos acceder al gestor a través de un navegado apuntando a la IP de la máquina.



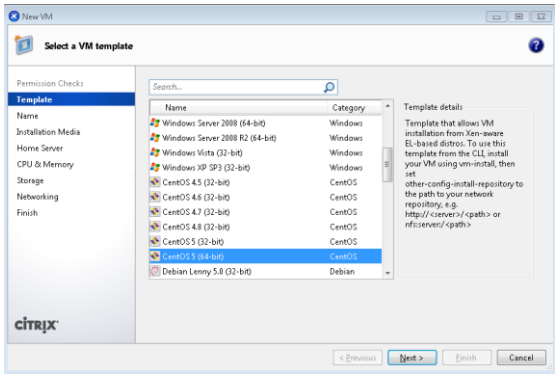
A raíz de estas imágenes, se pueden vez las semejanzas entre los dos entornos, pues en definitiva, están basados en la misma raíz.

4.3.6 Creación de Máquinas Virtuales

La creación de una nueva máquina virtual a través de cualquiera de las interfaces de gestión es un proceso muy sencillo. Se trata básicamente de seguir una serie de sencillos pasos y completar la información relacionada con la versión de sistema operativo a instalar y los detalles sobre los recursos de hardware virtual que se desean dotar a la máquina virtual.

A continuación, se describe cómo se puede realizar esta operación a través de la interfaz XenCenter de Citrix.

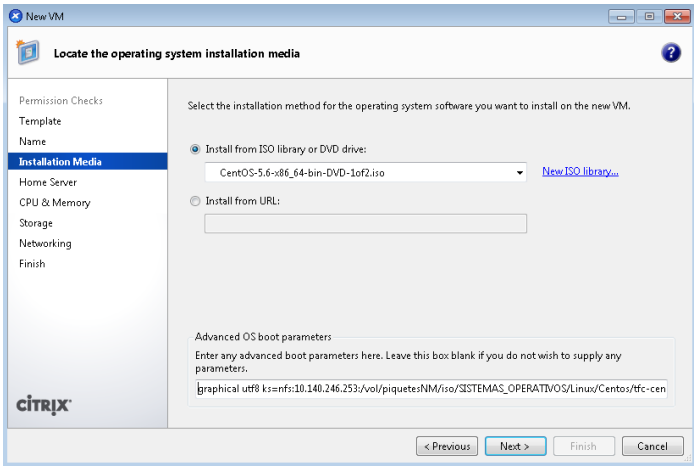
Se comenzará seleccionando la opción de creación de nueva máquina virtual disponible en la barra de herramientas superior.



Aparecerá un formulario que permite la selección del Sistema Operativo que se desea instalar.

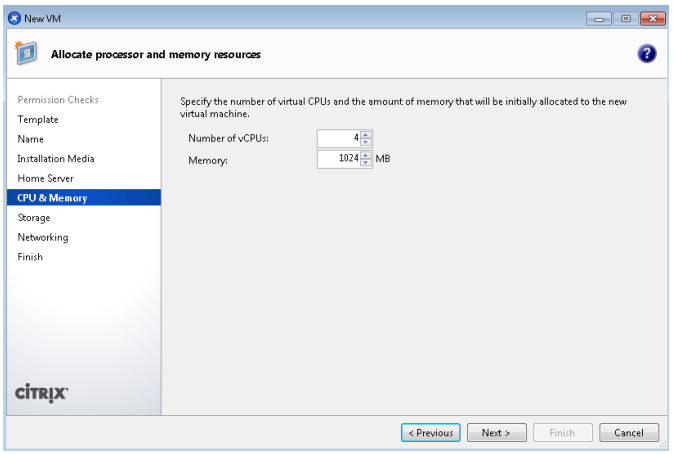
Seleccionado este, y tras pulsar sobre Next, se podrá indicar el nombre que ha de tener la máquina virtual y que la identificará. Posteriormente, se solicitará el medio origen de la instalación. En este caso, se podrá seleccionar o bien una ISO del repositorio que se ha configurado o bien se indicará una URL

(http/ftp) en la que estén ubicadas las mismas. En este caso, se opta por indicar la imagen ISO del repositorio.



Además, en las opciones avanzadas, se ha añadido un parámetro que nos permite apuntar la instalación a un fichero de kickstart preconfigurado y que automatizará el proceso de instalación con nuestras necesidades. Este fichero puede estar accesible por múltiples protocolos (http/ftp/nfs). En esta instalación se hace uso de NFS.

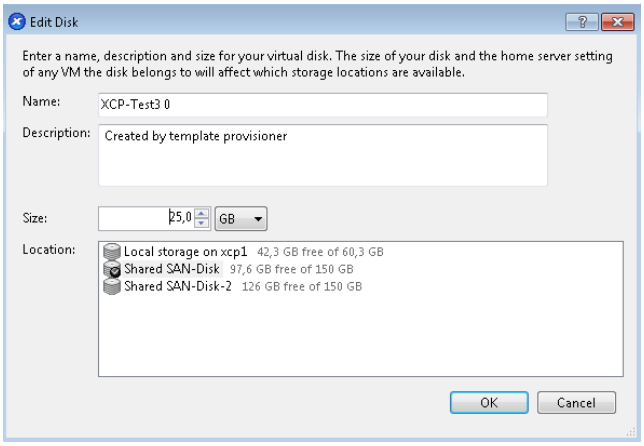
Se continúa con la configuración de los valores hardware de la nueva máquina virtual. Se podrá seleccionar el número de CPU virtuales y memoria con las que arrancará inicialmente el servidor.



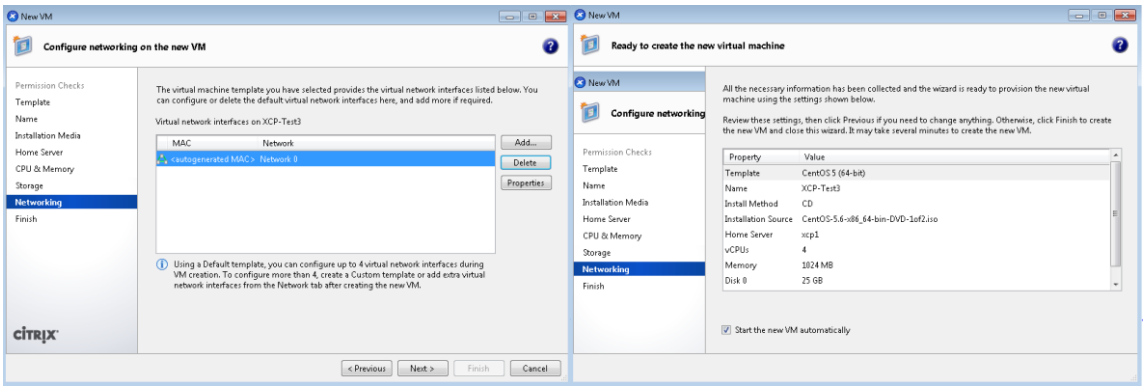
Después, se solicita introducir los valores relativos al disco virtual. Su nombre, ubicación y tamaño.

Para facilitar esta tarea, el sistema nos muestra los dispositivos de almacenamiento configurados y el espacio disponible para la creación de nuevos discos virtuales.

Se introduce el tamaño, y se continúa con el asistente.



Como paso final, se pregunta por el número de interfaces de red que se generarán en la máquina virtual, así como a qué dispositivo se asociará de aquellos creado en el Pool del servidor.



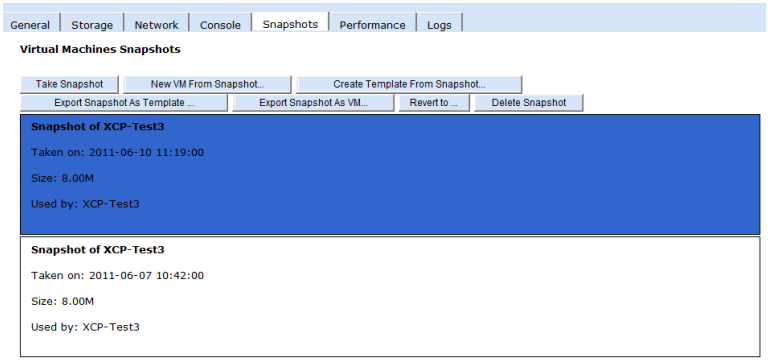
Tras todos estos pasos, habrá finalizado la configuración de la máquina virtual. Automáticamente se arrancará (si se selecciona dicha opción) y comenzará la instalación del sistema operativo.

4.3.7 Clonado de Máquinas Virtuales

Aunque la instalación de máquinas virtuales se puede automatizar enormemente, la funcionalidad de clonarlas supone un gran ahorro de tiempo. Especialmente si hay tareas de configuración de aplicaciones que no pueden ser automatizadas en tiempo de instalación (o que suponen un gran esfuerzo).

Las herramientas que dispone XCP a este propósito son muy potentes, sencillas y rápidas. Se disponen de varias alternativas para esta tarea.

- Snapshots. Se puede crear una foto de cualquiera de las máquinas virtuales y a raíz de dicha foto, crear una nueva máquina virtual.
- Copiar. Esta opción es el clonado como tal, pues permite copiar o mover una máquina virtual (previamente apagada) con otro nombre.
- Desde una plantilla previamente definida. Esta es otra opción que nos permite convertir una instalación nueva en plantilla para ser utilizada para futuras instalaciones.



4.4 Plantilla KickStart de CentOS

La instalación de las máquinas virtuales que se han utilizado en esta memoria se ha realizado haciendo uso de la siguiente plantilla de kickstart. A través de este formato, es posible automatizar las instalaciones de los sistemas operativos basados en la distribución Linux RedHat (como es el caso de CentOS). En ella se especifican las configuraciones de red y servicios, así como los requisitos mínimos de paquetería para que se puedan ejecutar las pruebas de rendimiento necesarias para la evaluación de los entornos virtuales.

```
install
reboot
text
nfs --server <DIRECCION SERVIDOR NFS> --dir <RUTA AL DIRECTORIO DE LAS
ISO>
lang en_US.UTF-8
keyboard es
network --device eth0 --bootproto dhcp --hostname XCP-Test2
rootpw --iscrypted $1$nkwd4zDJ$Jsi.02VcGvvV1zPKN/Khp/
firewall --enabled --port=22:tcp
authconfig --enableshadow --enablemd5
selinux --enforcing
timezone --utc Atlantic/Canary
bootloader --location=mbr --driveorder=xvda
clearpart --all --drives=xvda --initlabel
part /boot --fstype ext3 --size=100 --ondisk=xvda
part pv.2 --size=0 --grow --ondisk=xvda
volgroup VolGroup00 --pesize=32768 pv.2
logvol / --fstype ext3 --name=LogVol00 --vgname=VolGroup00 --size=1024
--grow
logvol swap --fstype swap --name=LogVol01 --vgname=VolGroup00 --
size=1008 --grow --maxsize=2016

%packages
@base
@core
@dialup
@editors
@ftp-server
@legacy-network-server
@mail-server
@network-server
@server-cfg
@text-internet
keyutils
iscsi-initiator-utils
trousers
fipscheck
device-mapper-multipath
gcc
gcc-c++
make
autoconf
automake
php53*
java-openjdk
gcc-gfortran
zlib-devel
numactl-devel
blas
```

```
blas-devel
lapack-devel
boost-devel
bzip2-devel
SDL-devel
openmpi-devel
openmpi
libaio-devel
```

4.5 Instalación/Configuración de Phoronix Test Suite

Una vez instalada una máquina virtual con los requisitos mínimos para la utilización del paquete de ejecución de pruebas de rendimiento, nos descargaremos el producto de su página principal (http://www.phoronix-test-suite.com/?k=downloads).

Los principales prerrequisitos que se han instalado están descritos en la plantilla de instalación de kickstart utilizada para el despliegue de las máquinas virtuales.

Para este proyecto, se ha utilizado la última versión disponible en el momento, concretamente Phoronix Test Suite 3.0.1 (http://www.phoronix-test-suite.com/download.php?file=phoronix-test-suite-3.0.1).

Descargado el paquete phoronix-test-suite-3.0.1.tar.gz, se descomprimirá, y se ejecuta el script de instalación que se encuentra en el raíz del nuevo directorio que se ha creado. Una vez finalizada la ejecución, el software se habrá instalado en las ubicaciones por defecto.

```
# ./install-sh
./install-sh: line 83: xdg-mime: command not found
./install-sh: line 84: xdg-mime: command not found
./install-sh: line 85: xdg-icon-resource: command not found

Phoronix Test Suite Installation Completed

Executable File: /usr/bin/phoronix-test-suite
Documentation: /usr/share/doc/phoronix-test-suite/
Phoronix Test Suite Files: /usr/share/phoronix-test-suite/
```

Los errores que se indican corresponden a la falta de entorno gráfico en la máquina utilizada para la instalación.

Es importante realizar estos pasos como usuario root, pues así dispondremos de del producto para todos los usuarios de la máquina. En caso necesario, se podría instalar por un usuario no privilegiado, siempre y cuando se indique la ruta donde se instalarán los binarios de la distribución.

En este punto, disponemos de las bases para ejecutar perfiles de pruebas de rendimiento. Sin embargo, estos no vienen por defecto en el paquete del producto, y serán descargados de forma automática cuando procedamos a ejecutar las pruebas. Además, el software será capaz de identificar las dependencias de paquetería (para algunas distribuciones Linux) e instalarlas en caso de necesitarlas.

Los comandos principales para interactuar con el software son:

- **install**: Instalar nuevos perfiles de pruebas en el sistema.
- **benchmark**: ejecuta un perfil de pruebas sobre el sistema.
- **upload-result**: permite subir a la página openbenchmarking.co el resultado de alguna de las pruebas realizadas.
- **network-setup**: permite modificar la configuración de acceso a internet de la aplicación. En esta instalación ha sido necesario para ajustar el acceso a través de un servidor proxy.

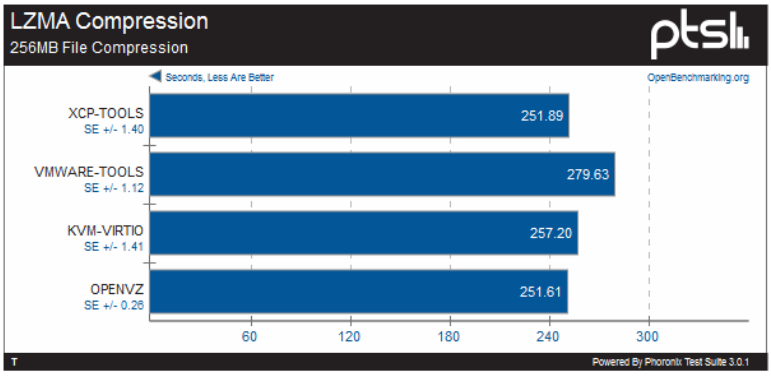
De forma general, una vez se comience a utilizar la aplicación, se generará un directorio en el home del usuario denominado .phoronix-test-suite. En este directorio se almacenarán los perfiles de pruebas y los resultados de la ejecución de los mismos. Estos resultados son visibles a través de un navegador, pues los almacena en formato html.

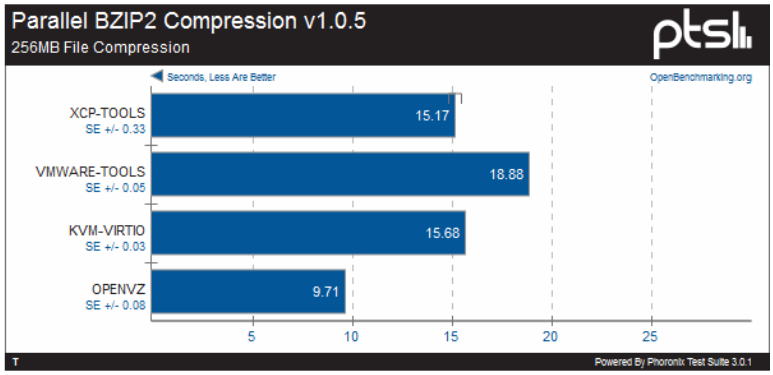
Se tiene la oportunidad de enlazar los resultados con la web de openbenchmarking.com y acceder a los mismos a través de ella. Con lo que se facilita la tarea de analizar y contrastar la información entre sí y entre otras plataformas que han subido los usuarios.

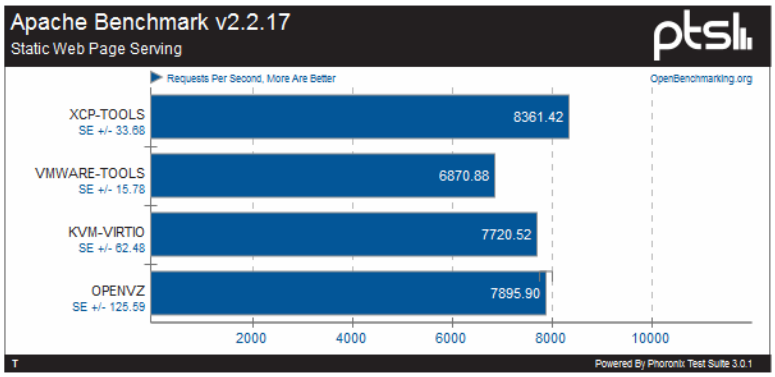
4.6 Resultados completos de las pruebas de rendimiento

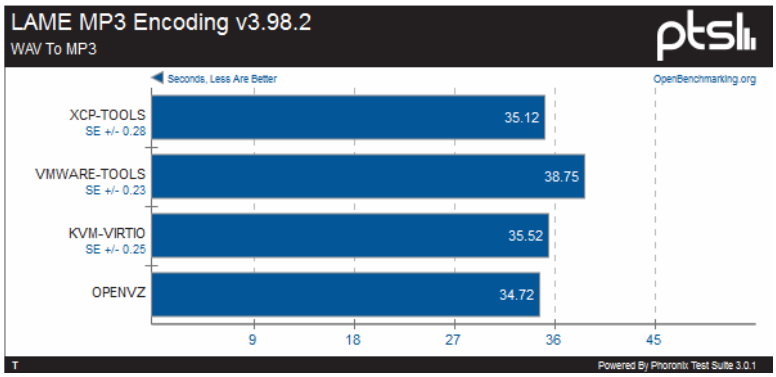
4.6.1 pts/linux-system

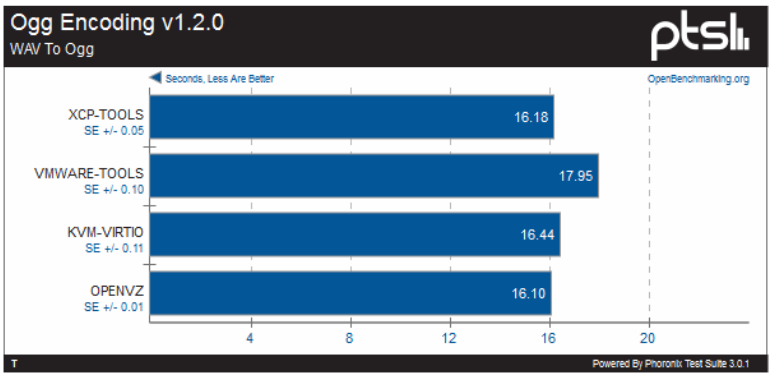
Phoronix Test Suite	XCP-TOOLS	VMWARE-TOOLS	KVM-VIRTIO	OPENVZ
LZMA Compression	251.89	279.63	257.20	251.61
Parallel BZIP2 Compression	15.17	18.88	15.68	9.71
Apache Benchmark	8361.42	6870.88	7720.52	7895.90
LAME MP3 Encoding	35.12	38.75	35.52	34.72
Ogg Encoding	16.18	17.95	16.44	16.10
FFmpeg	19.20	21.59	19.71	18.66
GnuPG	19.78	22.18		19.36
GMPbench	5389.90	5079.50		5415.75
C-Ray	91.02	96.54	91.55	47.36
POV-Ray	1369	1525	1402	1365
Timed MAFFT Alignment	36.17	40.25	37.04	36.08
Timed HMMer Search	33.52	37.32	33.47	20.97
Threaded I/O Tester	184.49	249.02	177.71	27.84
Threaded I/O Tester	1004.16	824.69	3300.58	2868.17
PostMark	1086	1263	1136	543
Dbench	166.70	288.48	239.40	17.41
GraphicsMagick	86	73	80	96
GraphicsMagick	34	30	31	49
OpenSSL	38.85	36.68	38.30	38.95
Crafty	748.08	832.98	764.75	749.22
Sunflow Rendering System	7.61	7.83	7.46	4.49
Sudokut	40.79	46.19	42.74	37.90
dcraw	35.86	39.90	37.86	35.40
Minion	186.21	210.12	191.57	
PyBench	4065	4367	4133	4011

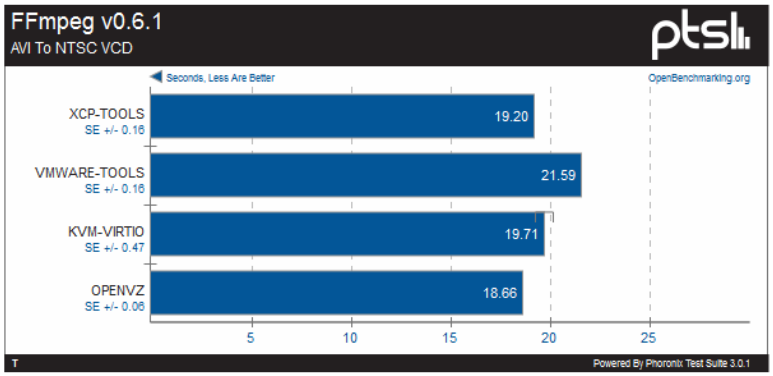


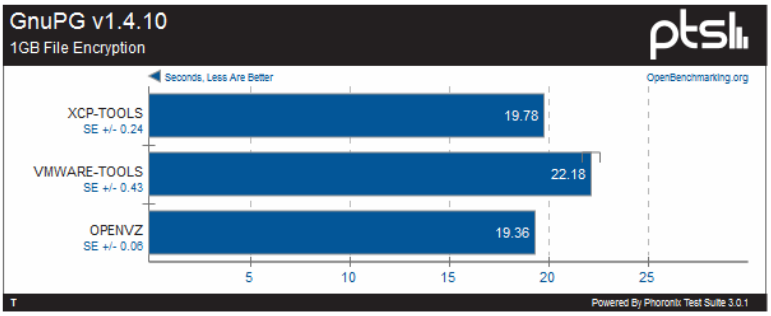


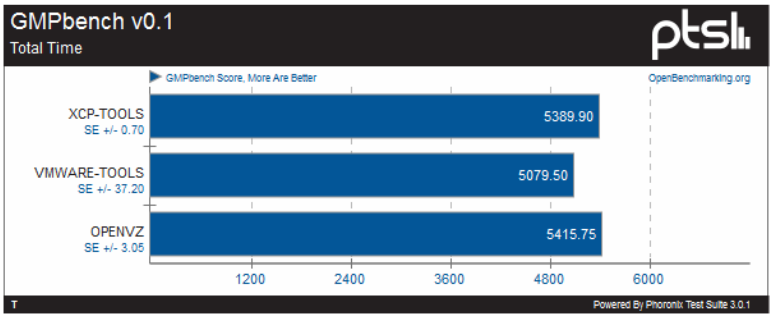

Apache Benchmark v2.2.17
Static Web Page Serving
Requests Per Second, More Are Better
OpenBenchmarking.org
XCP-TOOLS
SE +/- 33.68
8361.42
VMWARE-TOOLS
SE +/- 15.78
6870.88
KVM-VIRTIO
SE +/- 62.48
7720.52
OPENVZ
SE +/- 125.59
7895.90
2000
4000
6000
8000
10000
Powered By Phoronix Test Suite 3.0.1

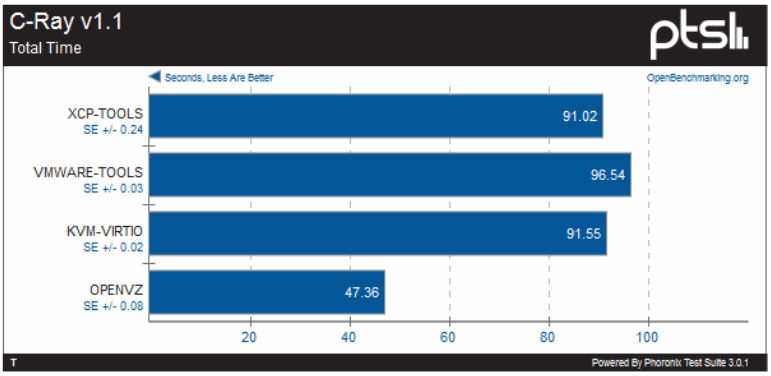

LAME MP3 Encoding v3.98.2
WAV To MP3
Seconds, Less Are Better
OpenBenchmarking.org
XCP-TOOLS
SE +/- 0.28
35.12
VMWARE-TOOLS
SE +/- 0.23
38.75
KVM-VIRTIO
SE +/- 0.25
35.52
OPENVZ
34.72
9
18
27
36
45
Powered By Phoronix Test Suite 3.0.1

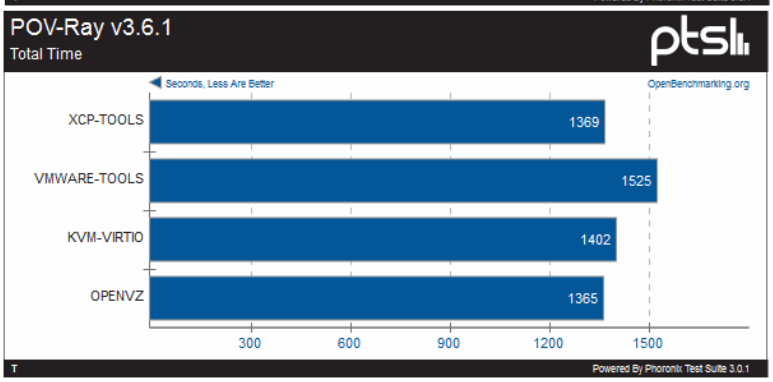

Ogg Encoding v1.2.0
WAV To Ogg
Seconds, Less Are Better
OpenBenchmarking.org
XCP-TOOLS
SE +/- 0.05
16.18
VMWARE-TOOLS
SE +/- 0.10
17.95
KVM-VIRTIO
SE +/- 0.11
16.44
OPENVZ
SE +/- 0.01
16.10
4
8
12
16
20
Powered By Phoronix Test Suite 3.0.1

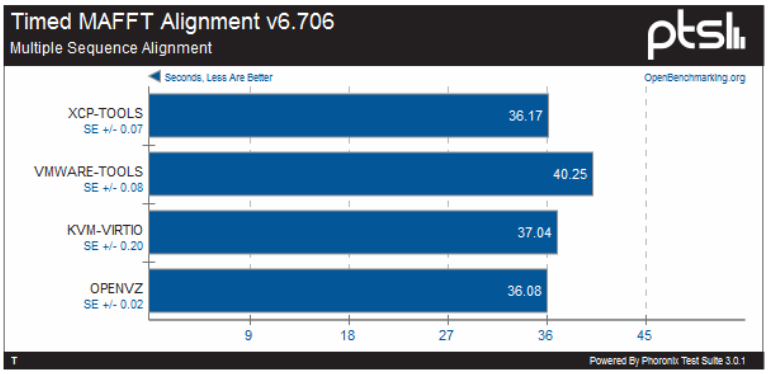

FFmpeg v0.6.1
AVI To NTSC VCD
Seconds, Less Are Better
OpenBenchmarking.org
XCP-TOOLS
SE +/- 0.16
19.20
VMWARE-TOOLS
SE +/- 0.16
21.59
KVM-VIRTIO
SE +/- 0.47
19.71
OPENVZ
SE +/- 0.06
18.66
5
10
15
20
25
Powered By Phoronix Test Suite 3.0.1

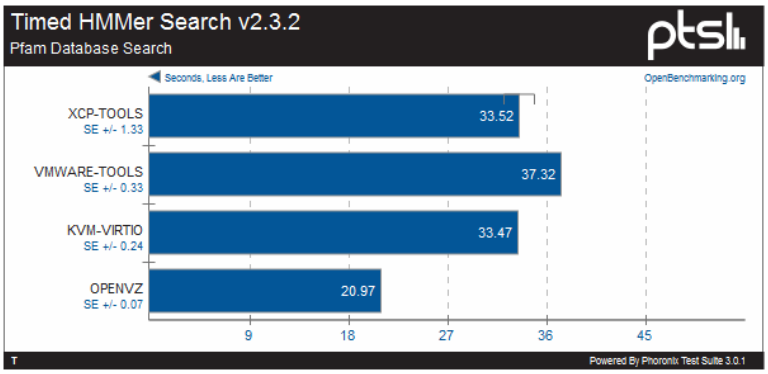

GnuPG v1.4.10
1GB File Encryption
ptsh
Seconds, Less Are Better
OpenBenchmarking.org
XCP-TOOLS
SE +/- 0.24
19.78
VMWARE-TOOLS
SE +/- 0.43
22.18
OPENVZ
SE +/- 0.06
19.36
5
10
15
20
25
Powered By Phoronix Test Suite 3.0.1

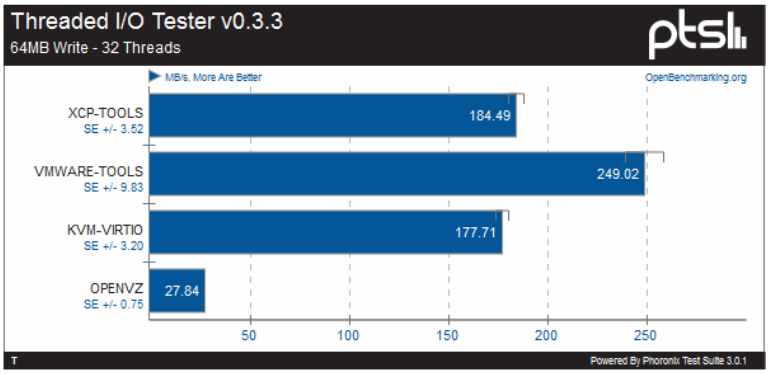

GMPbench v0.1
Total Time
ptsh
GMPbench Score, More Are Better
OpenBenchmarking.org
XCP-TOOLS
SE +/- 0.70
5389.90
VMWARE-TOOLS
SE +/- 37.20
5079.50
OPENVZ
SE +/- 3.05
5415.75
1200
2400
3600
4800
6000
Powered By Phoronix Test Suite 3.0.1

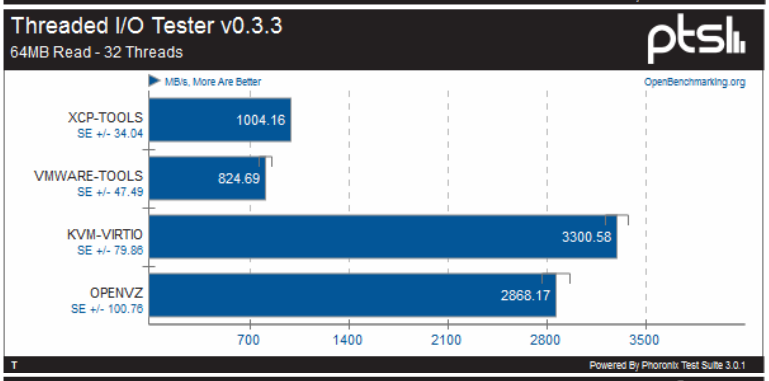

C-Ray v1.1
Total Time
ptsh
Seconds, Less Are Better
OpenBenchmarking.org
XCP-TOOLS
SE +/- 0.24
91.02
VMWARE-TOOLS
SE +/- 0.03
96.54
KVM-VIRTIO
SE +/- 0.02
91.55
OPENVZ
SE +/- 0.08
47.36
20
40
60
80
100
Powered By Phoronix Test Suite 3.0.1

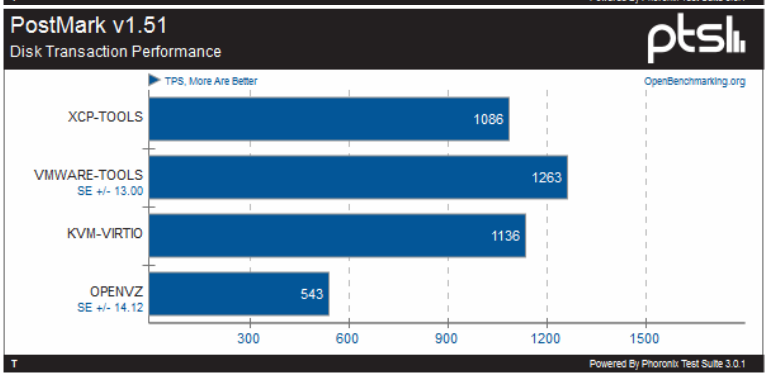

POV-Ray v3.6.1
Total Time
ptsh
Seconds, Less Are Better
OpenBenchmarking.org
XCP-TOOLS
1369
VMWARE-TOOLS
1525
KVM-VIRTIO
1402
OPENVZ
1365
300
600
900
1200
1500
Powered By Phoronix Test Suite 3.0.1

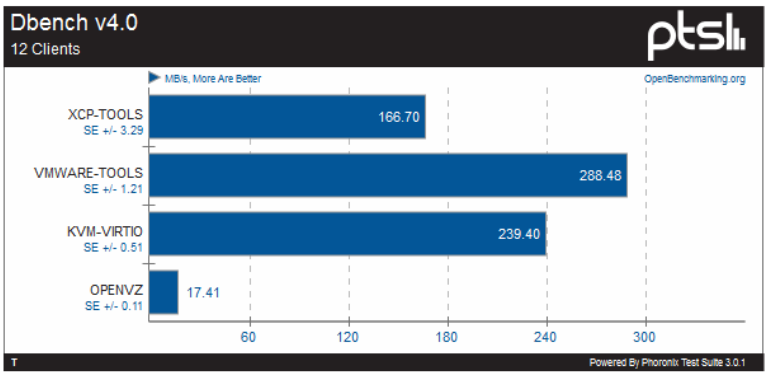

Timed MAFFT Alignment v6.706
Multiple Sequence Alignment
ptsh
Seconds, Less Are Better
OpenBenchmarking.org
XCP-TOOLS
SE +/- 0.07
36.17
VMWARE-TOOLS
SE +/- 0.08
40.25
KVM-VIRTIO
SE +/- 0.20
37.04
OPENVZ
SE +/- 0.02
36.08
9
18
27
36
45
Powered By Phoronix Test Suite 3.0.1

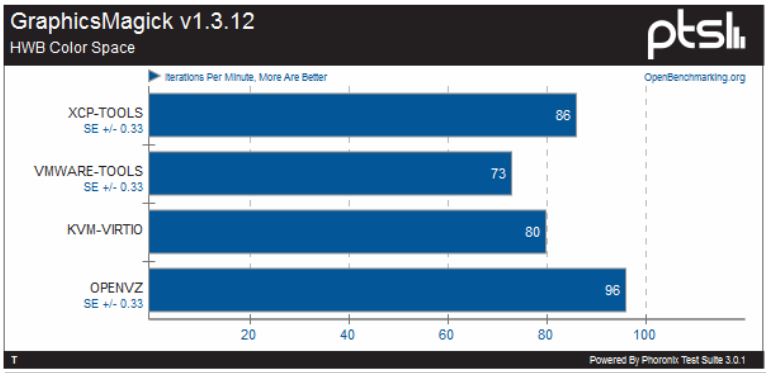

Timed HMMer Search v2.3.2
Pfam Database Search
Seconds, Less Are Better
OpenBenchmarking.org
XCP-TOOLS
SE +/- 1.33
33.52
VMWARE-TOOLS
SE +/- 0.33
37.32
KVM-VIRTIO
SE +/- 0.24
33.47
OPENVZ
SE +/- 0.07
20.97
9
18
27
36
45
Powered By Phoronix Test Suite 3.0.1

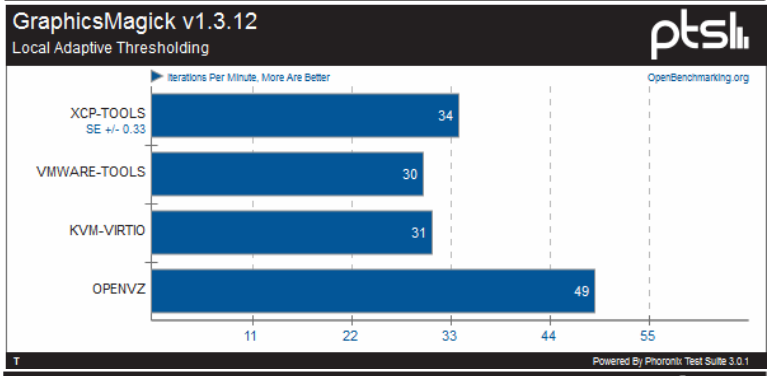

Threaded I/O Tester v0.3.3
64MB Write - 32 Threads
MB/s, More Are Better
OpenBenchmarking.org
XCP-TOOLS
SE +/- 3.52
184.49
VMWARE-TOOLS
SE +/- 9.83
249.02
KVM-VIRTIO
SE +/- 3.20
177.71
OPENVZ
SE +/- 0.75
27.84
50
100
150
200
250
Powered By Phoronix Test Suite 3.0.1

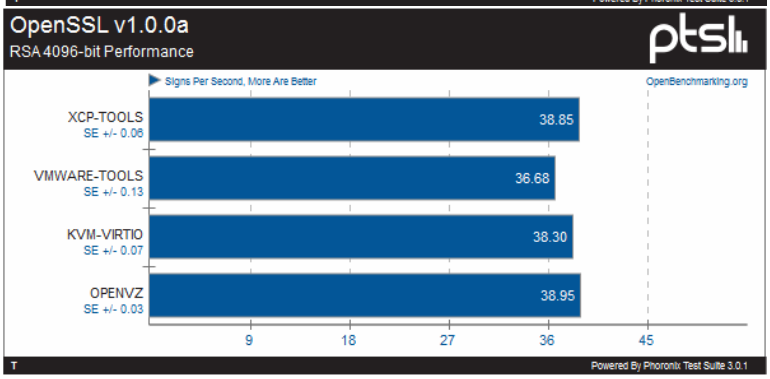

Threaded I/O Tester v0.3.3
64MB Read - 32 Threads
MB/s, More Are Better
OpenBenchmarking.org
XCP-TOOLS
SE +/- 34.04
1004.16
VMWARE-TOOLS
SE +/- 47.49
824.69
KVM-VIRTIO
SE +/- 79.86
3300.58
OPENVZ
SE +/- 100.76
2868.17
700
1400
2100
2800
3500
Powered By Phoronix Test Suite 3.0.1

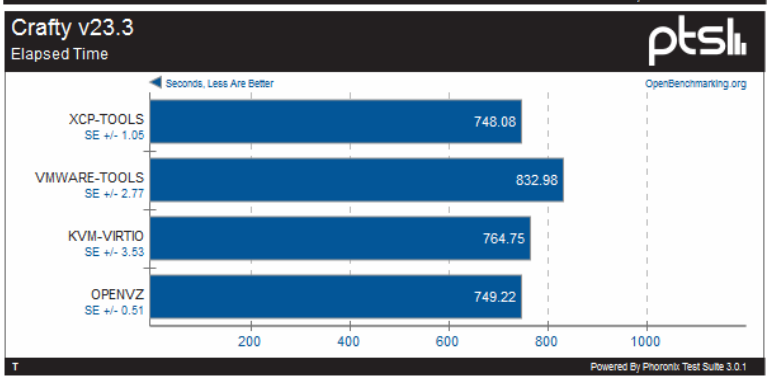

PostMark v1.51
Disk Transaction Performance
TPS, More Are Better
OpenBenchmarking.org
XCP-TOOLS
1086
VMWARE-TOOLS
SE +/- 13.00
1263
KVM-VIRTIO
1136
OPENVZ
SE +/- 14.12
543
300
600
900
1200
1500
Powered By Phoronix Test Suite 3.0.1

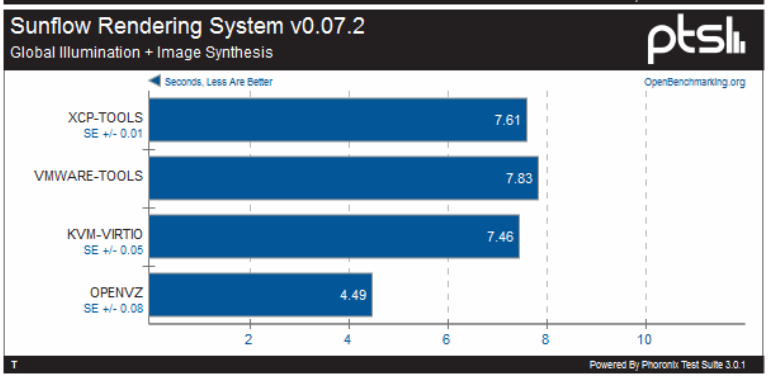

Dbench v4.0
12 Clients
MB/s, More Are Better
OpenBenchmarking.org
XCP-TOOLS
SE +/- 3.29
166.70
VMWARE-TOOLS
SE +/- 1.21
288.48
KVM-VIRTIO
SE +/- 0.51
239.40
OPENVZ
SE +/- 0.11
17.41
60
120
180
240
300
Powered By Phoronix Test Suite 3.0.1

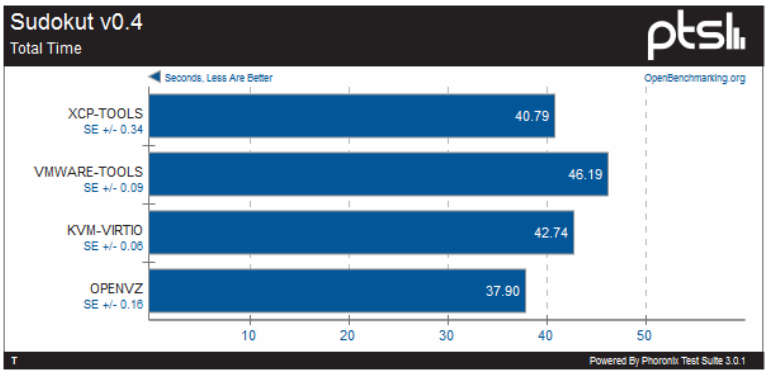

GraphicsMagick v1.3.12
HWB Color Space
ptsl
Iterations Per Minute, More Are Better
OpenBenchmarking.org
XCP-TOOLS
SE +/- 0.33
86
VMWARE-TOOLS
SE +/- 0.33
73
KVM-VIRTIO
80
OPENVZ
SE +/- 0.33
96
20
40
60
80
100
Powered By Phoronix Test Suite 3.0.1

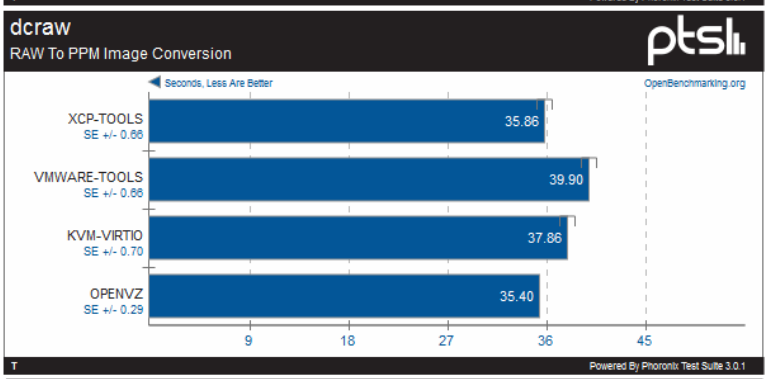

GraphicsMagick v1.3.12
Local Adaptive Thresholding
ptsl
Iterations Per Minute, More Are Better
OpenBenchmarking.org
XCP-TOOLS
SE +/- 0.33
34
VMWARE-TOOLS
30
KVM-VIRTIO
31
OPENVZ
49
11
22
33
44
55
Powered By Phoronix Test Suite 3.0.1

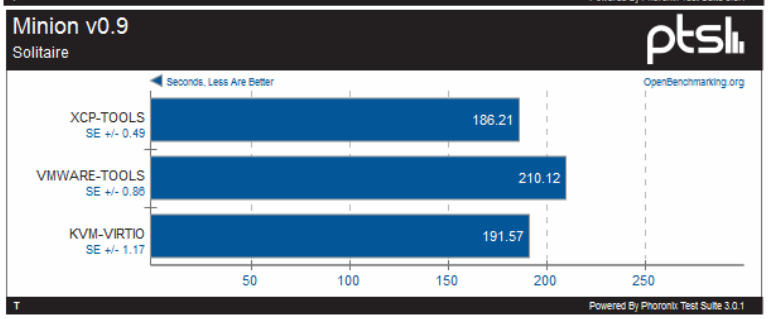

OpenSSL v1.0.0a
RSA 4096-bit Performance
ptsl
Signs Per Second, More Are Better
OpenBenchmarking.org
XCP-TOOLS
SE +/- 0.06
38.85
VMWARE-TOOLS
SE +/- 0.13
36.68
KVM-VIRTIO
SE +/- 0.07
38.30
OPENVZ
SE +/- 0.03
38.95
9
18
27
36
45
Powered By Phoronix Test Suite 3.0.1

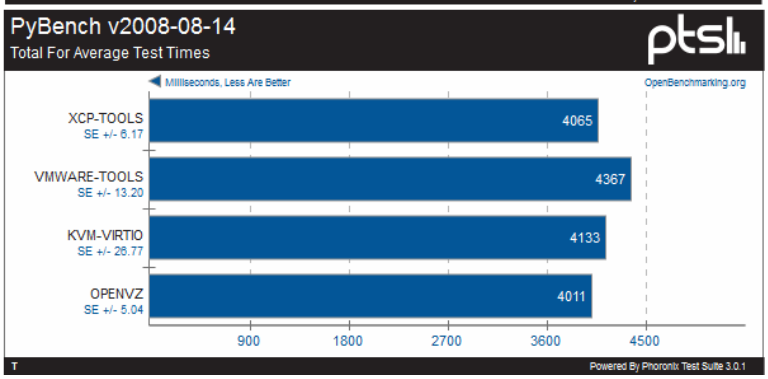

Crafty v23.3
Elapsed Time
ptsl
Seconds, Less Are Better
OpenBenchmarking.org
XCP-TOOLS
SE +/- 1.05
748.08
VMWARE-TOOLS
SE +/- 2.77
832.98
KVM-VIRTIO
SE +/- 3.53
764.75
OPENVZ
SE +/- 0.51
749.22
200
400
600
800
1000
Powered By Phoronix Test Suite 3.0.1


Sunflow Rendering System v0.07.2
Global Illumination + Image Synthesis
ptsl
Seconds, Less Are Better
OpenBenchmarking.org
XCP-TOOLS
SE +/- 0.01
7.61
VMWARE-TOOLS
7.83
KVM-VIRTIO
SE +/- 0.05
7.46
OPENVZ
SE +/- 0.08
4.49
2
4
6
8
10
Powered By Phoronix Test Suite 3.0.1







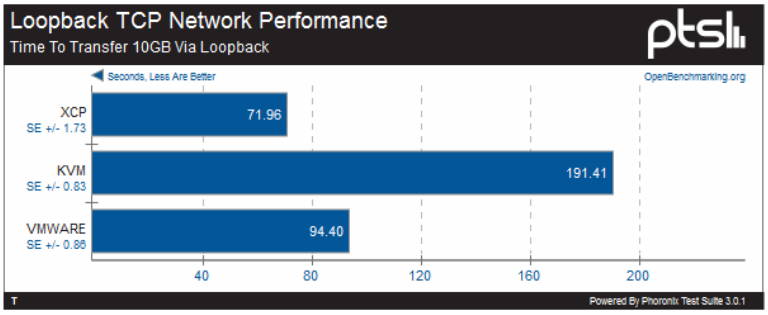


4.6.2 pts/network

xcp-network-centos56-run1

	XCP	KVM	VMWARE
Loopback TCP Network Performance	71.96	191.41	94.40
Difference	1.62x	1.00x	1.51x
Standard Error	1.73	0.83	0.86
Standard Deviation	5.89%	0.75%	1.57%

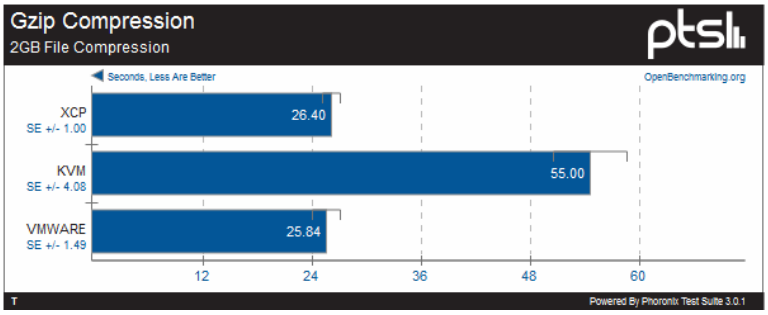
OpenBenchmar

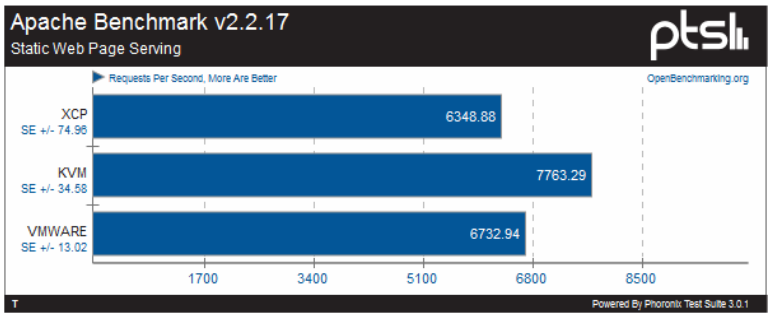


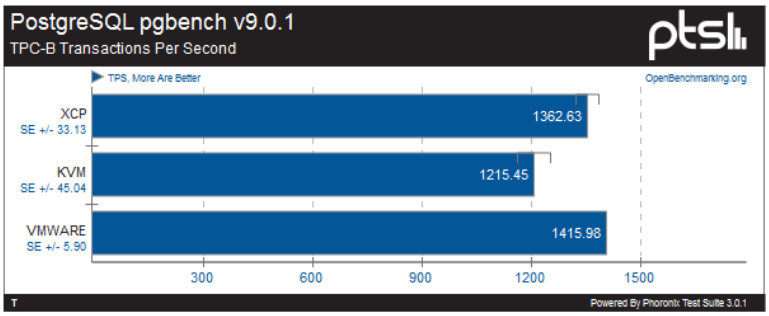
4.6.3 pts/disk

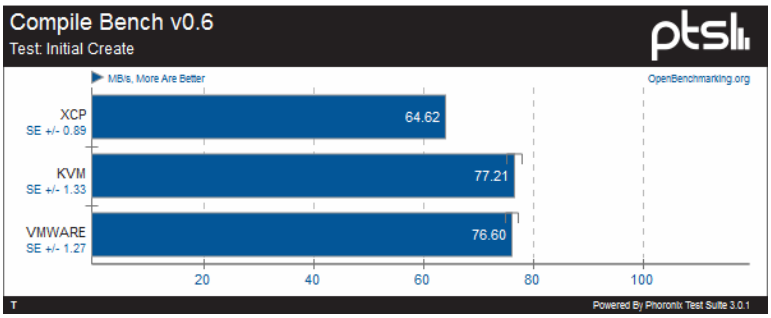
	XCP	KVM	VMWARE
Gzip Compression	26.40	55.00	25.84
Apache Benchmark	6348.88	7763.29	6732.94
PostgreSQL pgbench	1362.63	1215.45	1415.98
Compile Bench	64.62	77.21	76.60
IOzone	182.74	141.61	210.93
IOzone	46.16	48.50	186.96
Dbench	86.18	99.17	107.31
Dbench	155.70	222.98	294.90
Dbench	201.82	293.18	371.95
Dbench	222.38	296.61	378.92
FS-Mark	107.17	100.40	80.03
Flexible IO Tester	23.10	33.34	28.84
Threaded I/O Tester	260.25	170.74	366.15
Threaded I/O Tester	76.04	52.79	61.85
PostMark	1009	1096	1238
AIO-Stress	165.71	240.91	329.71
Unpacking The Linux Kernel	23.54	22.87	18.43

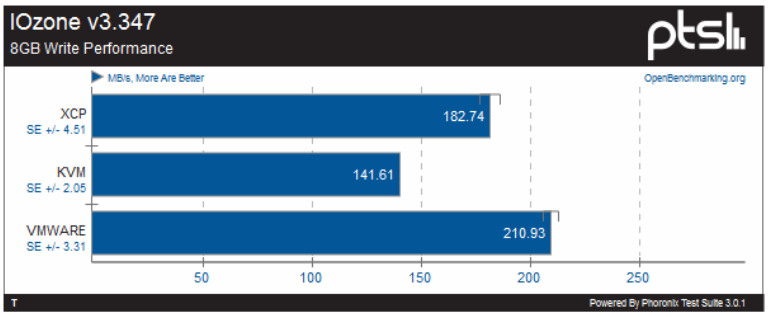
OpenBenchmarking.org

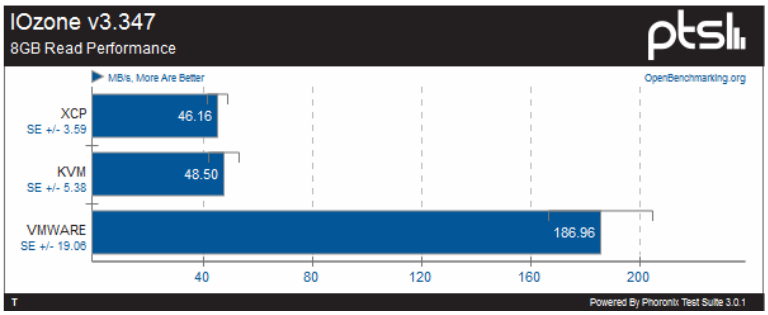


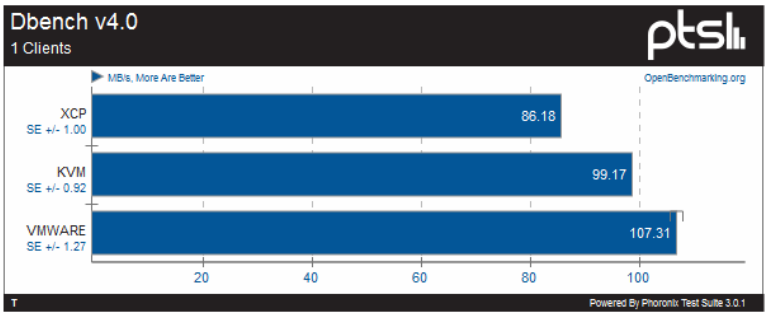


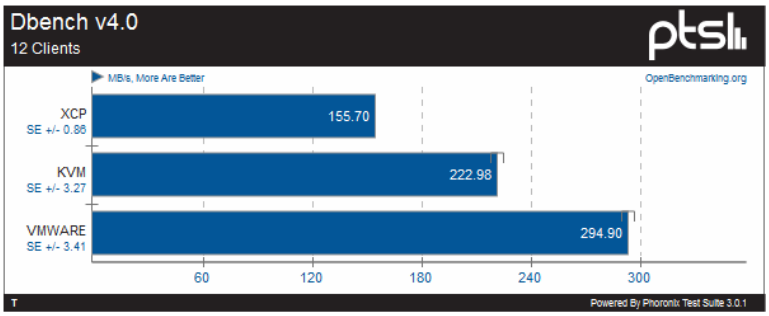

PostgreSQL pgbench v9.0.1
TPC-B Transactions Per Second
ptsl
TPS, More Are Better
OpenBenchmarking.org
XCP
SE +/- 33.13
1362.63
KVM
SE +/- 45.04
1215.45
VMWARE
SE +/- 5.90
1415.98
300
600
900
1200
1500
Powered By Phoronix Test Suite 3.0.1

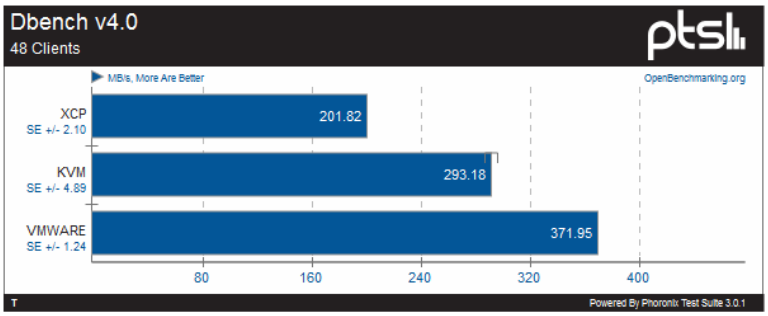

Compile Bench v0.6
Test: Initial Create
ptsl
MB/s, More Are Better
OpenBenchmarking.org
XCP
SE +/- 0.89
64.62
KVM
SE +/- 1.33
77.21
VMWARE
SE +/- 1.27
76.60
20
40
60
80
100
Powered By Phoronix Test Suite 3.0.1

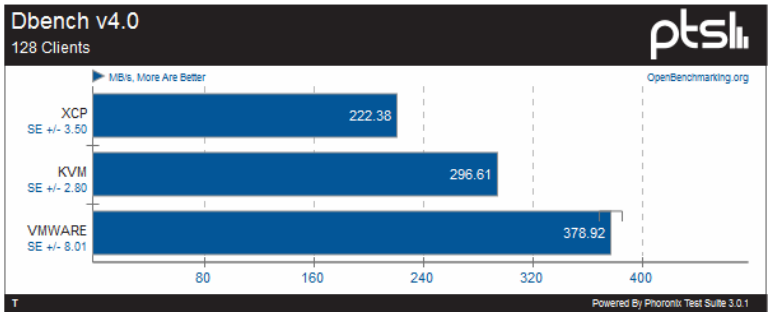

IOzone v3.347
8GB Write Performance
ptsl
MB/s, More Are Better
OpenBenchmarking.org
XCP
SE +/- 4.51
182.74
KVM
SE +/- 2.05
141.61
VMWARE
SE +/- 3.31
210.93
50
100
150
200
250
Powered By Phoronix Test Suite 3.0.1

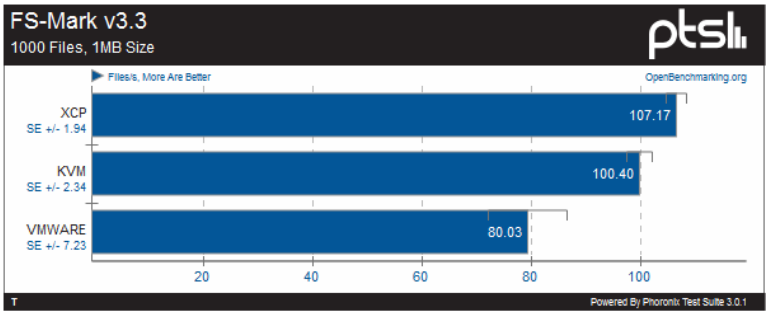

IOzone v3.347
8GB Read Performance
ptsl
MB/s, More Are Better
OpenBenchmarking.org
XCP
SE +/- 3.59
46.16
KVM
SE +/- 5.38
48.50
VMWARE
SE +/- 19.06
186.96
40
80
120
160
200
Powered By Phoronix Test Suite 3.0.1

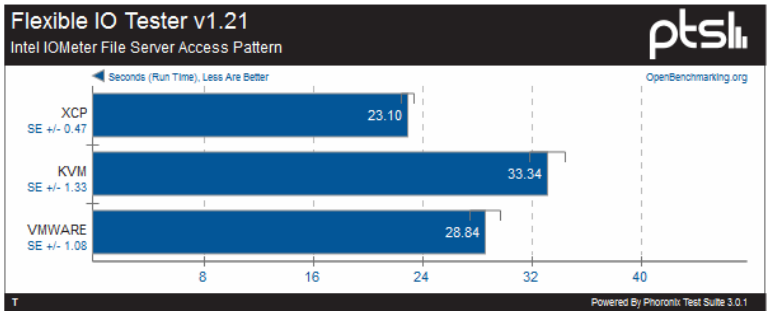

Dbench v4.0
1 Clients
ptsl
MB/s, More Are Better
OpenBenchmarking.org
XCP
SE +/- 1.00
86.18
KVM
SE +/- 0.92
99.17
VMWARE
SE +/- 1.27
107.31
20
40
60
80
100
Powered By Phoronix Test Suite 3.0.1

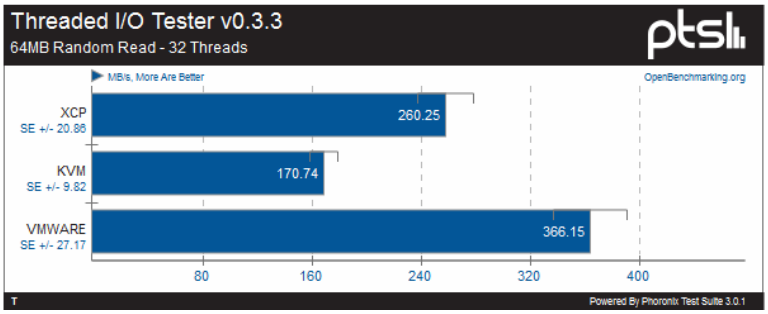

Dbench v4.0
12 Clients
MB/s, More Are Better
OpenBenchmarking.org
XCP
SE +/- 0.86
155.70
KVM
SE +/- 3.27
222.98
VMWARE
SE +/- 3.41
294.90
60
120
180
240
300
Powered By Phoronix Test Suite 3.0.1

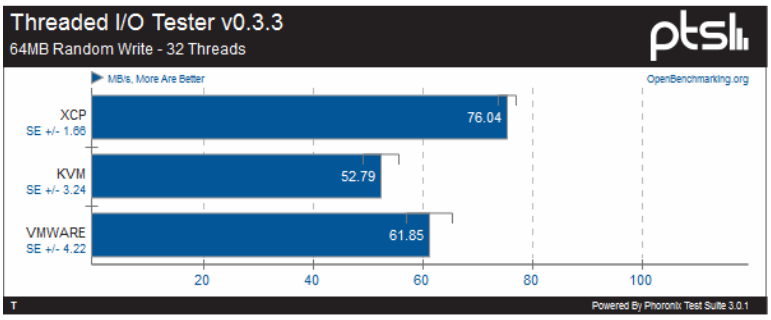

Dbench v4.0
48 Clients
MB/s, More Are Better
OpenBenchmarking.org
XCP
SE +/- 2.10
201.82
KVM
SE +/- 4.89
293.18
VMWARE
SE +/- 1.24
371.95
80
160
240
320
400
Powered By Phoronix Test Suite 3.0.1

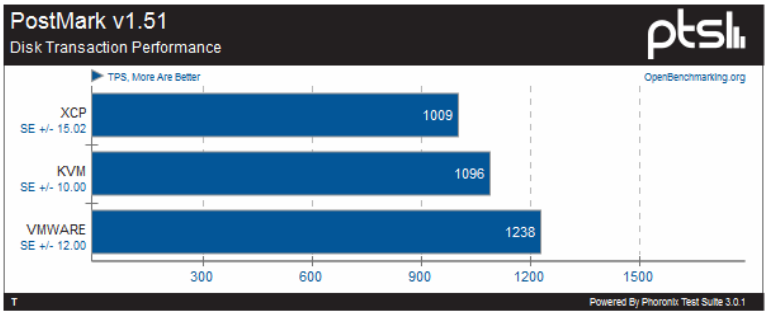

Dbench v4.0
128 Clients
MB/s, More Are Better
OpenBenchmarking.org
XCP
SE +/- 3.50
222.38
KVM
SE +/- 2.80
296.61
VMWARE
SE +/- 8.01
378.92
80
160
240
320
400
Powered By Phoronix Test Suite 3.0.1

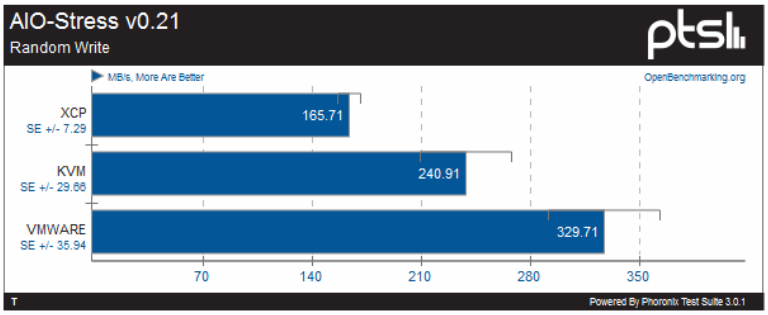

FS-Mark v3.3
1000 Files, 1MB Size
Files/s, More Are Better
OpenBenchmarking.org
XCP
SE +/- 1.94
107.17
KVM
SE +/- 2.34
100.40
VMWARE
SE +/- 7.23
80.03
20
40
60
80
100
Powered By Phoronix Test Suite 3.0.1

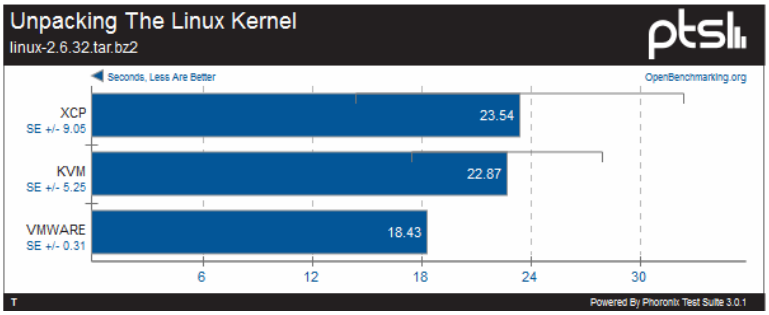

Flexible IO Tester v1.21
Intel IOMeter File Server Access Pattern
Seconds (Run Time), Less Are Better
OpenBenchmarking.org
XCP
SE +/- 0.47
23.10
KVM
SE +/- 1.33
33.34
VMWARE
SE +/- 1.08
28.84
8
16
24
32
40
Powered By Phoronix Test Suite 3.0.1


Threaded I/O Tester v0.3.3
64MB Random Read - 32 Threads
ptsl
MB/s, More Are Better
OpenBenchmarking.org
XCP
SE +/- 20.86
260.25
KVM
SE +/- 9.82
170.74
VMWARE
SE +/- 27.17
366.15
80
160
240
320
400
Powered By Phoronix Test Suite 3.0.1


Threaded I/O Tester v0.3.3
64MB Random Write - 32 Threads
ptsl
MB/s, More Are Better
OpenBenchmarking.org
XCP
SE +/- 1.66
76.04
KVM
SE +/- 3.24
52.79
VMWARE
SE +/- 4.22
61.85
20
40
60
80
100
Powered By Phoronix Test Suite 3.0.1


PostMark v1.51
Disk Transaction Performance
ptsl
TPS, More Are Better
OpenBenchmarking.org
XCP
SE +/- 15.02
1009
KVM
SE +/- 10.00
1096
VMWARE
SE +/- 12.00
1238
300
600
900
1200
1500
Powered By Phoronix Test Suite 3.0.1


AIO-Stress v0.21
Random Write
ptsl
MB/s, More Are Better
OpenBenchmarking.org
XCP
SE +/- 7.29
165.71
KVM
SE +/- 29.66
240.91
VMWARE
SE +/- 35.94
329.71
70
140
210
280
350
Powered By Phoronix Test Suite 3.0.1

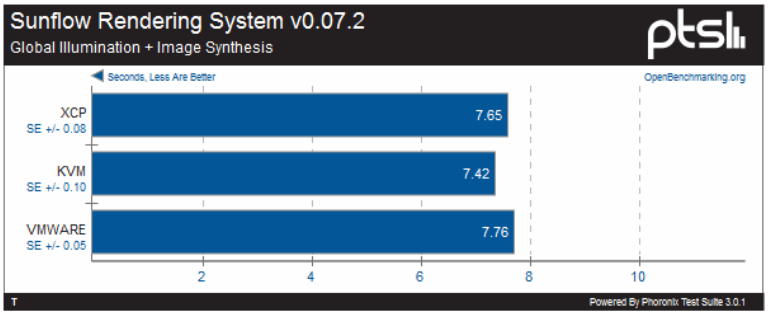

Unpacking The Linux Kernel
linux-2.6.32.tar.bz2
ptsl
Seconds, Less Are Better
OpenBenchmarking.org
XCP
SE +/- 9.05
23.54
KVM
SE +/- 5.25
22.87
VMWARE
SE +/- 0.31
18.43
6
12
18
24
30
Powered By Phoronix Test Suite 3.0.1

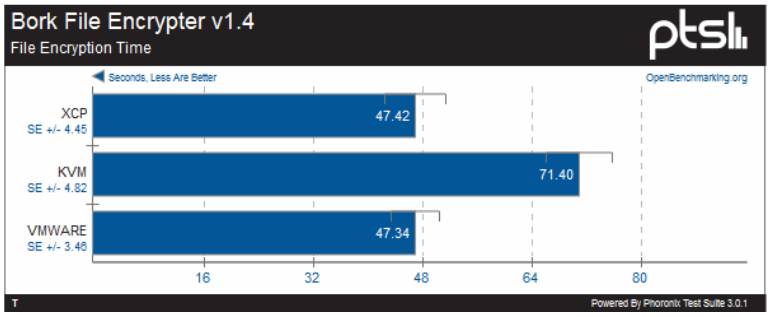
4.6.4 pts/java

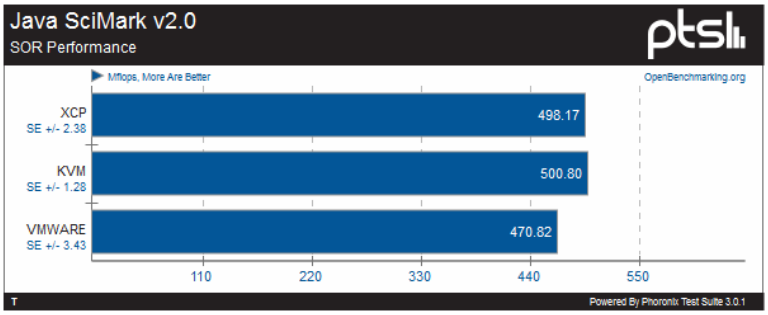
xcp-java-centos56-run1

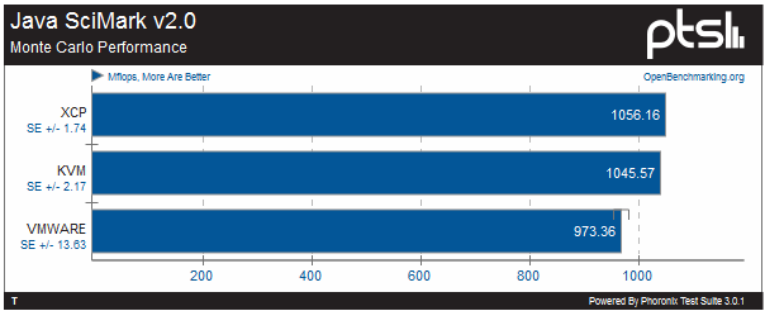
	XCP	KVM	VMWARE
Sunflow Rendering System	7.65	7.42	7.76
Bork File Encrypter	47.42	71.40	47.34
Java SciMark	498.17	500.80	470.82
Java SciMark	1056.16	1045.57	973.36
Java SciMark	850.40	844.07	792.68

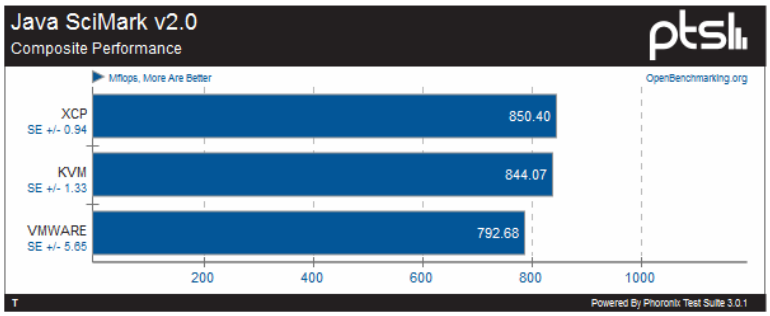
OpenBenchmarking.org









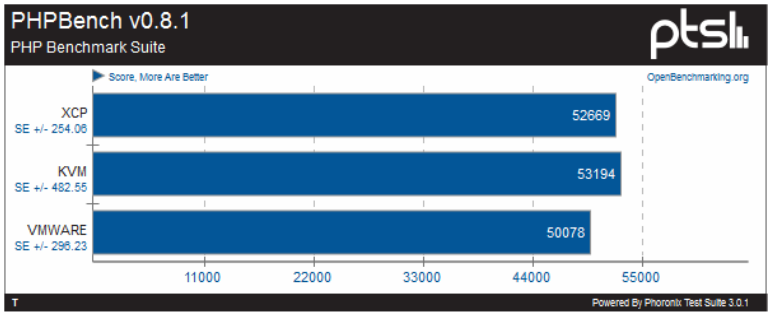


4.6.5 pts/phpbench

xcp-phpbench-centos56-run1

	XCP	KVM	VMWARE
PHPBench	52669	53194	50078
Difference	1.05x	1.06x	1.00x
Standard Error	254.06	482.55	296.23
Standard Deviation	0.84%	1.57%	1.02%

Open

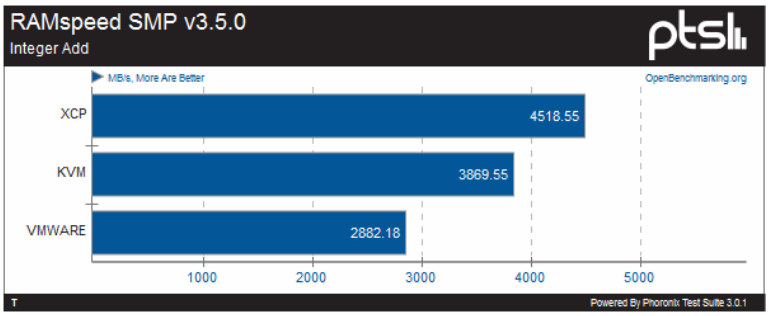


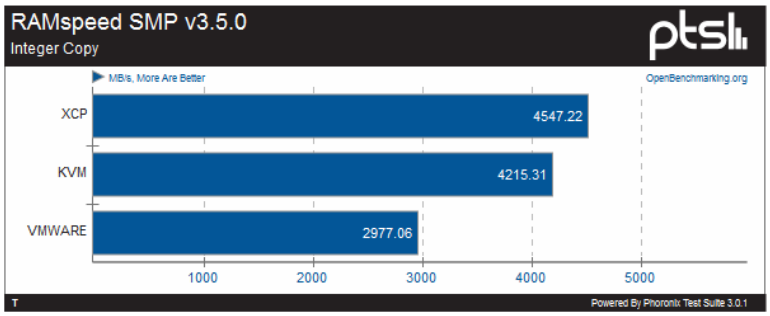
4.6.6 pts/memory

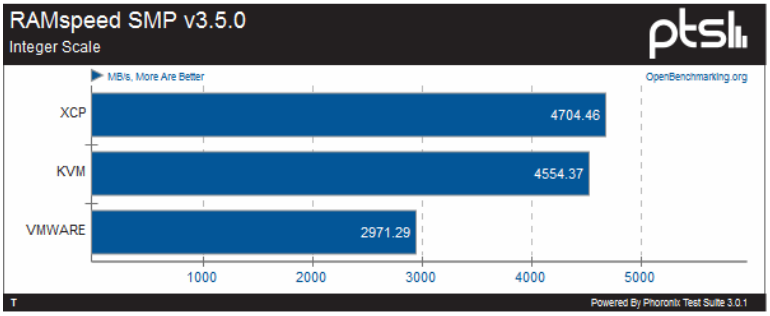
xcp-memory-centos56-run1

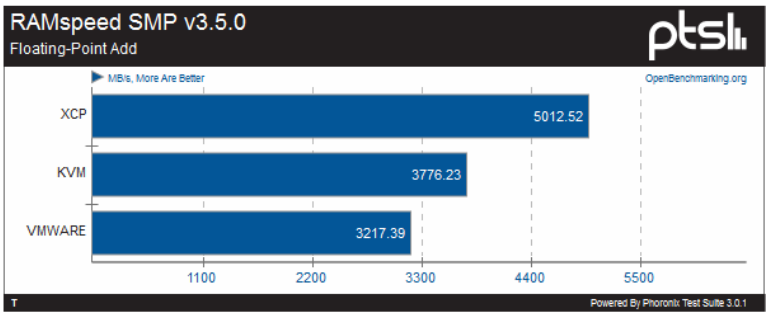
	XCP	KVM	VMWARE
RAMspeed SMP	4518.55	3869.55	2882.18
RAMspeed SMP	4547.22	4215.31	2977.06
RAMspeed SMP	4704.46	4554.37	2971.29
RAMspeed SMP	5012.52	3776.23	3217.39
Stream	5267.61	5719.29	3449.85
Stream	5124.35	5732.21	3214.03
Stream	4961.75	5435.73	3172.14
CacheBench	2241.99	2226.55	2099.21
CacheBench	9881.72	9801.82	9191.10

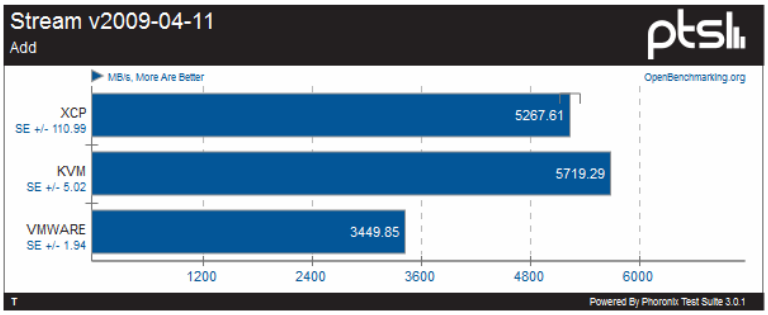
OpenBenchm

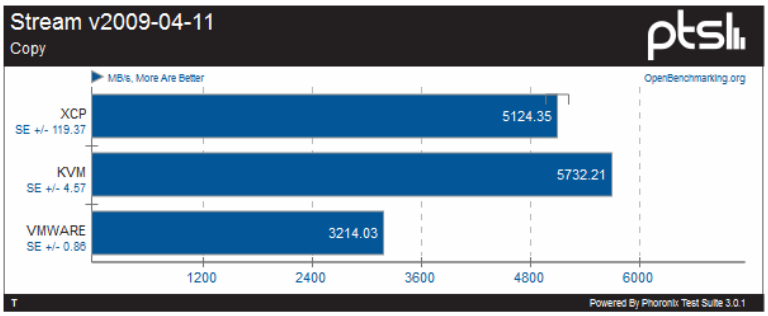


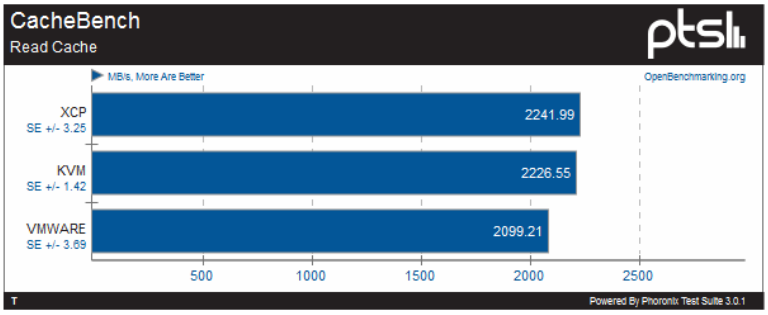

RAMspeed SMP v3.5.0
Integer Copy
ptsl
MB/s, More Are Better
OpenBenchmarking.org
XCP
4547.22
KVM
4215.31
VMWARE
2977.06
1000
2000
3000
4000
5000
Powered By Phoronix Test Suite 3.0.1

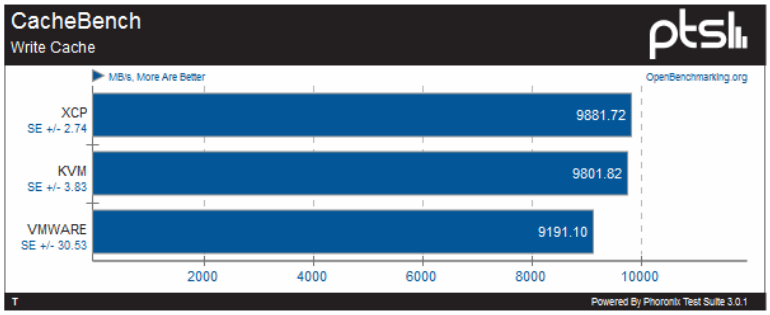

RAMspeed SMP v3.5.0
Integer Scale
ptsl
MB/s, More Are Better
OpenBenchmarking.org
XCP
4704.46
KVM
4554.37
VMWARE
2971.29
1000
2000
3000
4000
5000
Powered By Phoronix Test Suite 3.0.1


RAMspeed SMP v3.5.0
Floating-Point Add
ptsl
MB/s, More Are Better
OpenBenchmarking.org
XCP
5012.52
KVM
3776.23
VMWARE
3217.39
1100
2200
3300
4400
5500
Powered By Phoronix Test Suite 3.0.1


Stream v2009-04-11
Add
ptsl
MB/s, More Are Better
OpenBenchmarking.org
XCP
SE +/- 110.99
5267.61
KVM
SE +/- 5.02
5719.29
VMWARE
SE +/- 1.94
3449.85
1200
2400
3600
4800
6000
Powered By Phoronix Test Suite 3.0.1


Stream v2009-04-11
Copy
ptsl
MB/s, More Are Better
OpenBenchmarking.org
XCP
SE +/- 119.37
5124.35
KVM
SE +/- 4.57
5732.21
VMWARE
SE +/- 0.86
3214.03
1200
2400
3600
4800
6000
Powered By Phoronix Test Suite 3.0.1


CacheBench
Read Cache
ptsl
MB/s, More Are Better
OpenBenchmarking.org
XCP
SE +/- 3.25
2241.99
KVM
SE +/- 1.42
2226.55
VMWARE
SE +/- 3.69
2099.21
500
1000
1500
2000
2500
Powered By Phoronix Test Suite 3.0.1


CacheBench
Write Cache
ptsl
MB/s, More Are Better
OpenBenchmarking.org
XCP
SE +/- 2.74
9881.72
KVM
SE +/- 3.83
9801.82
VMWARE
SE +/- 30.53
9191.10
2000
4000
6000
8000
10000
Powered By Phoronix Test Suite 3.0.1