UNIVERSIDAD TÉCNICA PARTICULAR DE LOJA

La Universidad Católica de Loja

ÁREA TÉCNICA

TITULACIÓN DE INGENIERO EN SISTEMAS INFORMÁTICOS Y COMPUTACIÓN

Virtualización de servicios no críticos de la UTPL con plataformas Opensource.

TRABAJO DE FIN DE TITULACIÓN

AUTOR: Ordóñez Gonzalez, Juan Carlos

DIRECTORA: Torres Guarnizo, Diana Alexandra

LOJA - ECUADOR

2014

APROBACIÓN DEL DIRECTOR DEL TRABAJO DE FIN DE TITULACIÓN

Ingeniera
Diana Alexandra Torres Guarnizo
DOCENTE DE TITULACIÓN

De mi consideración:

Que el presente trabajo de investigación, *Virtualización de servicios no críticos de la UTPL con plataformas Opensource*, realizado por *Juan Carlos Ordoñez Gonzalez*, ha sido orientado y revisado durante su ejecución, por cuanto se aprueba la presentación del mismo.

Loja, septiembre del 2014.

_____

Ing. Diana Alexandra Torres Guarnizo

DECLARACIÓN DE AUTORÍA Y CESIÓN DE DERECHOS

“Yo Juan Carlos Ordóñez Gonzalez declaro ser autor (a) del presente trabajo de fin de titulación Virtualización de servicios no críticos de la UTPL con plataformas Opensource, de la Titulación Ingeniero en Sistemas Informáticos y Computación, siendo Diana Alexandra Torres Guarnizo, director (a) del presente trabajo; y eximo expresamente a la Universidad Técnica Particular de Loja y a sus representantes legales de posibles reclamos o acciones legales. Además certifico que las ideas, conceptos, procedimientos y resultados vertidos en el presente trabajo investigativo, son de mi exclusiva responsabilidad.

Adicionalmente declaro conocer y aceptar la disposición del Art. 67 del Estatuto Orgánico de la Universidad Técnica Particular de Loja que en su parte pertinente textualmente dice: “Forman parte del patrimonio de la Universidad la propiedad intelectual de investigaciones, trabajos científicos o técnicos y tesis de grado que se realicen a través, o con el apoyo financiero, académico o institucional (operativo) de la Universidad.

f. .....

Autor: Juan Carlos Ordoñez G.
Cédula: 1103989925

AGRADECIMIENTO

Agradezco en primer lugar a Dios, por permitirme culminar el presente proyecto de tesis.

A mi familia por el gran apoyo incondicional en todos los sentidos, a mis tutores, Ing(s) Diana y Alexander quienes influyeron con su continua valoración, empujando y animando moralmente en algunos tramos difíciles; dedicando tiempo y esfuerzo para la culminación exitosa del presente proyecto.

Finalmente a mi Universidad por las enseñanzas adquiridas, las experiencias y metas logradas pero sobre todo por la oportunidad de conocer buenos amigos que supieron enriquecerme y compartir más que académicamente, a nivel vivencial y personal.

DEDICATORIA

Este trabajo está totalmente dedicado a mi familia, a mis queridos padres Augustita y Luis, a mis entrañables hermanos Elisa, Javier, Maria Augusta y a mi amada esposa Katty quien siempre estuvo apoyándome en todo momento. De manera especial está dedicado a mi querida madre Augustita, quien por siempre será el gran modelo de persona a seguir, muy alegre, luchadora y desde hace poco un angel que nos guía desde el cielo. Para ella mi gran admiración, gratitud y para quienes siempre agradecemos a Dios el sentir y apreciar la entrega incondicional en cada una de las cosas que mi madre realizaba.

Añado a mis estimados amigos y amigas que a lo largo de la carrera universitaria hemos logrado superar obstáculos, incentivadonos y apoyandonos uno al otro, para lograr todos los objetivos propuestos, superando cada momento difícil que nos ha tocado vivir, así como también celebrando los éxitos y metas alcanzadas.

INDICE DE CONTENIDOS

ÍNDICE DE TABLAS

ÍNDICE DE FIGURAS

RESUMEN

El presente proyecto de fin de carrera determina como la virtualización, en especial la de servidores usando plataformas de código abierto, aplicada a los servicios que se clasificaron como menos críticos de la Universidad Técnica Particular de Loja, permite lograr varios beneficios expresados en baja de costes al reducir gastos de infraestructura física, energética y de mantenimiento operativo simplificando la administración usando Xen Open Source como plataformas centralizadas.

PALABRAS CLAVE: virtualización, código abierto, xen, servicios, servidores, utpl

ABSTRACT

This project analyzes especially as virtualization of servers using the open source platforms applied to services that were classified as less critical of the Universidad Técnica Particular de Loja achieves several benefits expressed in lower costs by reducing physical infrastructure costs, energy and operational maintenance using simplifying management and centralized Xen Open Source platforms.

KEYWORDS: virtualization, opensource, xen, services, servers, utpl.

INTRODUCCIÓN

Las empresas y organizaciones en general están siempre en busca de una continua evolución y desarrollo en sus servicios que les permitan extender su valor agregado frente al resto de competidores. Esto genera un aumento en la demanda de nueva tecnología, herramientas, aplicaciones y consecuentemente nueva infraestructura en equipos hardware. Sin darse cuenta, estas organizaciones experimentan un crecimiento descontrolado tanto en el campo de servidores como en aplicaciones, en donde se exige cada vez más recursos y por ende aumentan los costos de la infraestructura de hardware, electricidad y recursos humanos para su gestión. Es por esto que la UTPL, no ajena a esta realidad busca implementar soluciones prácticas a estos problemas, sin que se cause retraso o estancamiento en su continua labor de brindar servicios educativos y generar investigación de calidad.

El presente proyecto se realiza enfocado al estudio de las diferentes plataformas relacionadas con el campo de la *virtualización de servidores*, definiendo una plataforma idónea para el entorno de la UTPL, empleando para este fin exclusivamente software y herramientas Open Source. Todo lo antes mencionado está encaminado a realizar un análisis y estudio detallado, identificando los amplios beneficios que actualmente conlleva el usar *virtualización*, siendo implementado en algunos de los servicios clasificados como no críticos. En este campo denominado *virtualización de servicios* entran algunos conceptos, arquitecturas y tecnologías diferentes, pero todas se orientan hacia un mismo objetivo que es el de virtualizar y optimizar recursos. También nos enfocaremos en la búsqueda de una solución específica que facilite la gestión de los distintos servicios seleccionados como no críticos, para que posteriormente sean operativos en el momento de ser implementada la solución o plataforma.

Actualmente en la UTPL existen una variedad de servicios que están operativos y que ayudan a que las actividades en la institución sean más efectivas. Algunos de estos tienen un mayor grado de importancia o criticidad que otros según el ámbito de negocio de la organización, que en este caso es el de servicios educacionales. El presente proyecto se concentrará en los servicios que tienen un grado de criticidad baja, como punto inicial para comenzar la planeación de la virtualización de equipos con sus respectivos servicios operativos.

Para dicha planeación lo primero es realizar un análisis para determinar la criticidad de los servicios que existen actualmente en la Universidad Técnica Particular de Loja. En esta parte se establecen las variables que influyen sobre la criticidad y se cuantifican sus efectos. Se usa un método de clasificación propio, conforme a las necesidades de los servicios existentes en la UTPL, lo que conllevará a delimitar los resultados en varios niveles de criticidad, que finalmente se utilizan para establecer la estrategia al momento de seleccionar los más adecuados para la etapa de virtualización.

Finalmente después de realizar un monitoreo sobre cada uno de los equipos físicos y virtuales obtendremos algunos datos de rendimiento (entrada y salida de datos), que nos ayudarán a visualizar el rendimiento real que existe en los equipos, servicios y componentes virtualizados.

OBJETIVOS DEL PROYECTO

Objetivo General:

- Implementar una solución de virtualización OpenSource para los servicios clasificados como menos críticos en la UTPL.

Objetivos Específicos:

- Identificar los servicios menos críticos y seleccionar aquellos que son factibles virtualizar.
- Analizar las distintas soluciones en virtualización Open Source y seleccionar la más adecuada para adaptarla al entorno de la UTPL.
- Verificar el correcto rendimiento de la solución de virtualización en cuanto a procesamiento, uso de memoria RAM y flujo-entrada/salida- de tráfico de red en cada uno de los servicios.

CAPITULO I
ASPECTOS GENERALES DE LA VIRTUALIZACIÓN.

1.1 Antecedentes.

La virtualización tiene sus orígenes en los años sesenta, cuando se usaba las máquinas de tiempo compartido y la multiprogramación. Esta técnica tenía la propiedad de permitir que se desarrolle un programa en consola, mientras otro programador trabajaba en otra aplicación, evitando la espera entre procesos distintos.

Luego surgen supercomputadoras como la ATLAS y la IBM's M44/44x[1], las mismas que son una evolución de los inicios de la multiprogramación, tiempo compartido y el control de dispositivos compartidos, estos conceptos son enfocados como orígenes de la virtualización.

Especialmente la computadora IBM's M44 es la que introduce el término de máquina virtual al simular equipos virtuales 7044 (M44) o máquinas 44x usando memoria virtual compartida (en hardware) e implementando multiprogramación (en software).

Años después la compañía IBM junto al MIT[2], introducen el concepto de CTSS (siglas en inglés de Sistemas compatibles de tiempo compartido), bajo el estándar FMS (Fortran Monitor System), se comparte el trabajo por lotes en segundo plano con acceso a dispositivos y sin interrupción a los procesos iniciados en primera instancia.

Posteriormente se emprendieron varios proyectos y esfuerzos de virtualización hasta nuestros días, pero son demasiado numerosos para mencionarlos en su totalidad. Algunos han fallado mientras los otros han pasado a ser populares y sus tecnologías son aceptadas en toda la comunidad técnica.

Pero el verdadero surgimiento se da en los años de 1999 y 2000; cuando la mayoría de los esfuerzos estaban guiados tan solo en el campo de virtualización de servidores, se presentó la necesidad de simplificar los centros de datos en toda su amplitud por medio de

---

[1] "ATLAS: La primera de las supercomputadoras de la década de 1960, proyecto dirigido por el Departamento de Ingeniería Eléctrica en la Universidad de Manchester y financiado por Ferranty Limited, el Atlas fue el ordenador más rápido de aquel tiempo.

IBM M44/44x: Proyecto de supercomputador con arquitectura similar a la de Atlas liderado por IBM que consistía de una máquina de capacidad científica que simulaba máquinas virtuales serie 7044 y 44x." (Hoopes, 2009)

[2] MIT: Siglas de Massachusetts Institute of Technology.

virtualización. Es así que se toma otras áreas como el caso de: almacenamiento virtual, redes virtuales, equipos de escritorio virtuales y últimamente la virtualización de móviles y de aplicaciones.

Muchas compañías, como Sun, Microsoft, Citrix y VMware, han liberado productos de sus empresas, (ya sea por estrategia empresarial o por la competitividad existente); las mismas que tienen amplia aceptación en el mercado, atribuible en parte un numeroso grupo de consumidores, que han ganado hasta nuestros días.

1.2 Definición de Virtualización.

En términos generales a la *Virtualización* se la relaciona con el concepto de emular algo físico. Esta percepción no está lejos de la definición precisa, pero aplicada al campo computacional se define como el proceso de sustituir el hardware físico por software, el mismo que emula sus características y funcionamiento con el propósito de ofrecer una mayor flexibilidad y/o eficiencia en los recursos o dispositivos físicos. Aquí tenemos algunas definiciones de autores que se acercan bastante a esta temática:

> *"Es un marco o metodología de dividir los recursos de un computador hardware en ambientes de ejecución múltiples, aplicando una o más tecnologías como, el particionamiento de software o hardware, tiempo compartido, simulación parcial o completa, emulación, calidad del servicio y muchas otras."* (Williams, 2007)

> *"La virtualización se refiere a aquellas tecnologías diseñadas para ofrecer una capa de abstracción entre los sistemas de hardware y el software instalado en ellos".* (IDG, 2009)

1.3 Motivos del uso de la Virtualización.

El constante crecimiento de las aplicaciones y entornos de aplicación en las empresas y organizaciones, ha producido que la infraestructura de servidores empiece a crecer de forma exponencial. Esto claramente ha sido un factor preponderante para que la nueva

tendencia apunte hacia un enfoque de sistema centralizado, que junto al gran ahorro de recursos y costes han permitido que esta nueva tecnología llamada ***virtualización***, tenga el desarrollo y la aceptación que tienen actualmente y que por tanto sea considerada como una de las tecnologías emergentes.

> *"Gartner estima que aproximadamente el 90 por ciento del mercado de servidores está compuesto por servidores de arquitectura x86, pero basado en un modelo tradicional de una aplicación por servidor, aproximadamente el 80 al 90 por ciento de la capacidad de computación x86 no se utiliza en cualquier momento. La Virtualización promete liberar la mayor parte de esta capacidad subutilizada."* (Gartner, 2010)

Las ventajas mencionadas anteriormente son solo algunas de las razones por lo que se opta por una solución de virtualización. Existen otras ventajas según (Rule & Dirtnner, 2007)entre las que están:

- ✓ Reduce los tiempos de parada.
- ✓ Reducción de los costes de espacio y consumo de energía.
- ✓ Mejora de TCO (Costo total de operación) y ROI (Retorno de inversión).
- ✓ Migración en caliente de máquinas virtuales (live migration).
- ✓ Rápida incorporación de nuevos recursos para los servidores virtualizados.
- ✓ Reducción de los costes de infraestructura tecnológica gracias al aumento de la flexibilidad y eficiencia en el uso de recursos.
- ✓ Administración más simplificada y ahorro de tiempo en configuraciones innecesarias.
- ✓ Capacidad de clonación y copia de sistemas enteros.
- ✓ Propiedad de aislamiento: es decir un fallo general del sistema de una máquina virtual no afecta al resto de máquinas virtuales.

"Un entorno virtualizado requiere de un entorno de red confiable y de alta capacidad". (Rule & Dirtnner, 2007)

Para lograr robustecer las cargas de trabajo en un entorno virtualizado, es esencial que todos los componentes y subsistemas del servidor – tales como CPU, memoria, red y disco- puedan adecuar la carga de trabajo adicional. Si bien la mayoría de los productos de virtualización requiere una única conexión de red para funcionar, se debe dar una atención cuidadosa a la planificación de la infraestructura de red del entorno virtual y así se puede garantizar un rendimiento óptimo y de alta disponibilidad. Una buena práctica es contar con dispositivos redundantes para asegurar la continuidad del servicio. Esto ayuda de forma que ante una falla del dispositivo principal, entre a funcionamiento un dispositivo secundario de las mismas características, transparentemente con el mínimo impacto al equipo y por tanto al servicio.

El uso de varias máquinas virtuales sobre un equipo host aumentarán el tráfico de red, ya que por una misma interfaz física pasan los datos de diferentes hosts e interfaces de red virtuales. Así con varias cargas de trabajo, la capacidad de la red estará disminuida para satisfacer las necesidades de cada host virtual. Es por eso que siempre que el procesador anfitrión (Dom0), no se utilice al cien por ciento, el tráfico de red total será la suma del tráfico generado por cada máquina virtual.

1.3.1 Escenarios de uso.

Existe una variedad de escenarios que la virtualización tiene para lograr beneficios visibles, sobre todo contribuye a la mejora en eficiencia, eficacia y administración de la Infraestructura Tecnológica. La tendencia actual en proyectos de virtualización no solo busca ahorrar costes o utilizar al máximo la infraestructura de servidores, sino también se usa en planes de recuperación ante desastres, contingencia y continuidad de negocio basándose en sistemas de alta disponibilidad con el fin de precautelar el activo más valioso que es la información.

"Más del 80 por ciento de las empresas tienen ahora un programa o proyecto de virtualización, pero sólo el 25 por ciento de todas las cargas de trabajo estarán en una máquina virtual (VM) a fin de año 2010." (Gartner, 2010)

Entre los escenarios de uso más comunes tenemos:

- *"Consolidación de servidores: es muy común en las empresas tener servidores dedicados a ciertas aplicaciones que solamente utilizan una pequeña parte de los recursos que disponen. En estos casos se puede unificar las mismas en un servidor físico, dedicando una máquina virtual para cada aplicación, quedando las mismas aisladas entre sí.*
- *Independencia de hardware: cada sistema operativo virtualizado no tiene dependencia del hardware, ya que se crea una capa de abstracción entre el sistema operativo y el hardware real.*
- *Ambientes de prueba y desarrollo dinámicos: otro caso frecuente en las empresas es tener servidores dedicados a ambientes de desarrollo y prueba. Las máquinas virtuales permiten crear estos ambientes más rápidamente sin necesidad de depender de hardware adicional.*
- *Housing virtual: empresas proveedoras de servicios de hosting y housing pueden proveer servicios virtualizados, ahorrando costos y cobrando tarifas diferenciadas."* (Woitasen, 2006)

Además se añade el compromiso de que las organizaciones busquen que sus centros de procesamiento de datos (o CDPs), inserten tecnología que les ayude en la reducción de costes energéticos e incremento de la eficiencia operativa, junto con el compromiso por parte de la mayoría de países en *disminuir emanación de gases de*

efecto invernadero[3] a la atmósfera y el cumplimiento de las normativas de cada uno de los órganos reguladores respectivos.

"La Tecnología de Información y Comunicaciones de la industria es responsable de casi el 2% de las emisiones mundiales de CO2, la mayoría de los que resultan del consumo de energía de los PCs, servidores y sistemas de refrigeración." (Mingay, Simon, 2009)

"Los riesgos empresariales asociados a la contaminación son globales, de largo plazo e irreversibles. El Protocolo de Kioto[4] *limita a la mayoría de las compañías en su emisión de gases de invernadero."* (Lash, 2007)

1.4 Tipos de virtualización.

1.4.1 Virtualización a nivel de red

Entre los más conocidos tipos de virtualización a nivel de red están las VLANs, VIP y VPNs. Las primeras se refieren a las siglas en inglés de *redes de área local virtuales* y tienen la característica de poder crear redes lógicas independientes en una misma red física. Las segundas (*virtual IP*) hacen referencia a una dirección IP, que no está conectada con una computadora específica o tarjeta de interfaz de la red (NIC) en una computadora; los paquetes entrantes se envían a la dirección del VIP, pero todos los paquetes viajan a través de interfaces verdaderas de la red. Y por último tenemos las VPNs, cuyas siglas en inglés representan a *redes privadas virtuales* y su nombre se debe a que permiten establecer un canal de conexión segura entre dos puntos remotos y cuyos datos necesitan pasar por redes públicas, por ejemplo sobre internet.

[3]Efecto invernadero se refiere al proceso por el que ciertos gases de la atmósfera retienen gran parte de la radiación infrarroja emitida por la Tierra y la re-emiten de nuevo a la superficie terrestre calentando la misma.

[4]"El Protocolo de Kioto establece que los países desarrollados deben reducir sus emisiones de gases causantes del efecto invernadero en un 5,2% para el año 2012 respecto a las emisiones del año 1990. Sin embargo, este protocolo debe ser ratificado por al menos 55 países desarrollados cuyas emisiones de gases de efecto invernadero sumen entre sí el 55% del total."(Microsoft Encarta, 2009)

1.4.2 Virtualización a nivel de almacenamiento

El almacenamiento también tiene su campo abierto en la virtualización ya que a lo largo del tiempo se han creado soluciones muy efectivas para lograr ventajas como independencia de hardware y mayor flexibilidad, lo que nos permite expandir la infraestructura de almacenamiento de acuerdo con nuestras necesidades y posibilidades.

De la variedad de soluciones podemos destacar a la tecnología RAID y LVM[5]. A éstas se las puede considerar claramente como virtualización, ya que su funcionamiento básico radica en que todas las unidades que se utilizan y que interactúan, lo hacen como una unidad lógica única, aunque realmente estén compuestas por dos o más unidades físicas. Posteriormente aparecieron tecnologías más avanzadas de SAN[6] como iSCSI[7]que permiten una mejor gestión del almacenamiento, bajo los mismos principios de sus predecesores, pero con muchas más prestaciones como gestión avanzada de almacenamiento, migración de plataforma de almacenamiento en caliente sin tener ningún impacto para el host anfitrión.

1.4.3 Virtualización a nivel de aplicaciones.

Este es el enfoque más reciente de virtualización que ha surgido y que poco a poco va ganando más aceptación. La virtualización de aplicaciones utiliza componentes de software virtual para ejecutar las aplicaciones y datos, en lugar de realizar los procedimientos de instalación tradicionales. Los componentes de la aplicación pueden ser activados o desactivados al instante, ser reseteados y volver a su configuración predeterminada, y así

---

[5]RAID y LVM: Siglas de Redundant Array of Indepent Disks y Logical Volumes Management respectivamente. RAID es un sistema que usa múltiples discos duros para distribuir o replicar los datos a través de los discos duros que lo componen y LVM permite agrupar discos individuales en "grupos de volúmenes". Luego la capacidad de éstos grupos de volúmenes puede ser establecido en un volumen lógico, los cuales son accesados como dispositivos regulares. (López, 2009)

[6] SAN: es una red creada para enlazar equipos servidores, arrays de discos y equipos de respaldo, la cual está basada en tecnología fibre cannel (canal de fibra), o lo que es más comúnmente usado iSCSI.

[7] iSCI: Internet SCSI (Small Computer System Interface), iSCSI se utiliza para permitir el transporte de datos sobre redes IP locales o a través de largas distancias, obviamente siempre basándonos en alguna tecnología que permita el uso de IP, siempre hablamos en este caso de tecnologías de capa 2 tipo Ethernet, preferiblemente velocidades de por lo menos 1Gbps. (Collado, 2010)

mitigar el riesgo de interferencia con otro tipo de aplicaciones que se ejecutan en su propio espacio de cómputo.

1.4.4 Virtualización a nivel de servidores

Vamos a referirnos al tipo de virtualización que es más pertinente a esta investigación, ya que uno de nuestros objetivos es virtualizar equipos servidores con los servicios incluidos. Actualmente esta es la tecnología que más frecuentemente se usa y cuyas implementaciones existen para la mayoría de plataformas y arquitecturas de CPU. Aunque hay diferentes implementaciones y definiciones de un equipo virtual para arquitecturas x86-*las más genéricas*-, la clasificación más frecuente es la siguiente:

a) ***Emulación o simulación***

En este enfoque la máquina virtual emula un hardware completo, admitiendo un sistema operativo invitado o "guest" sin modificar, para una CPU completamente diferente pero con limitaciones ya que su desempeño es lento y por tanto no apto para entornos de producción. Entre algunos ejemplos de estas implementaciones tenemos a *QEMU y Bochs*[8].

b) ***Virtualización nativa o completa***

En este tipo de técnica los sistemas quedan sin modificar, además de que solo trabajan sobre la arquitectura del hardware real y por esto muchas instrucciones se ejecutan directamente en el hardware físico. Dicha virtualización llegó a ser posible gracias a la aparición de las extensiones de los fabricantes de los procesadores más comunes como son la tecnología INTEL-VT y AMD-V, ya que sin éstas no sería posible el virtualizar algunos sistemas propietarios.

[8] Bochs y Qemu se refieren a dos diferentes emuladores de plataforma x86 (32 bits). Son similares a las de VMware, ya que te dejan arrancar un ordenador virtual dentro de su sistema operativo normal.

c) Virtualización a nivel de sistema operativo.

Este esquema está basado en una simple instancia del sistema operativo, es decir no se virtualiza el hardware sino que más bien todos los procesos comparten un mismo kernel y los distintos procesos pertenecientes a cada servidor virtual se ejecutan aislados del resto.

"La ventaja de este enfoque es la separación de los procesos de usuario prácticamente sin pérdida en el rendimiento, pero al compartir todos los servidores virtuales el mismo kernel, no pueden obtenerse el resto de las ventajas de la virtualización." (VM Spain, 2010)

d) Paravirtualización.

La paravirtualización es una técnica que proporciona la simulación parcial de un determinado hardware. Aquí la mayoría de las características del equipo físico son simuladas y una de las más importantes es la gestión del espacio de virtualización; es decir la concesión de cada máquina virtual, de forma de que cada una se realiza en su propio espacio de dirección única. En general consiste en ejecutar sistemas operativos invitados (o guests) sobre otro sistema operativo que actúa como Hipervisor (host). Los guests tienen que comunicarse con el hipervisor para lograr la virtualización.

"Las ventajas de este enfoque radican en el buen rendimiento y la posibilidad de ejecutar distintos sistemas operativos como guests y su desventaja está en que los sistemas operativos guests deben ser modificados para funcionar en este tipo de esquema." (Smaldone, 2008)

Estos sistemas operativos paravirtualizados se denominan modificados, porque para lograr un funcionamiento óptimo entre el hipervisor y el sistema operativo son parcheados en su núcleo o kernel. Ésta modificación implica la creación de una capa más por encima del sistema operativo para la gestión de las instrucciones por parte del hipervisor; estopor supuesto sólo aplicable a sistemas Open Source. Dicha modificación es considerable, ya que se usa una técnica conocida como “ring deprivileging” que se detalla más adelante en el subcapítulo 1.5.1.

1.5 Hipervisor o Virtual machine monitor (VMM).

1.5.1 Definición de Hipervisor.

Un hipervisor se refiere a un programa que permite que varios sistemas operativos puedan compartir una gran cantidad de hardware. Cada uno parece tener en el anfitrión el conjunto de procesadores, memoria y otros recursos; sin embargo, es el hipervisor quién en realidad tiene el control del procesador central, los recursos, la asignación de lo que se necesita para cada sistema operativo y a su vez se asegura que los sistemas operativos invitados (llamados virtual machines) no se puedan interrumpir entre sí.

> *“La virtual machine manager (VMM) o hipervisor hace referencia a una tecnología que está compuesta por una capa de software que permite utilizar, al mismo tiempo, diferentes sistemas operativos o máquinas virtuales (sin modificar o modificados en el caso de paravirtualización) en una misma computadora central. Es decir es la parte principal de una máquina virtual que se encarga de manejar los recursos del sistema principal exportándolos a la máquina virtual.* (Talens-Oliag, Sergio, 2009)

Para esto el hipervisor utiliza una capa adicional que se ocupa de controlar la interacción entre los guest y mostrar los distintos sistemas operativos emplazados en la computadora real.

Generalmente en los equipos computacionales actuales, el sistema operativo es el software encargado de controlar los recursos de hardware como CPU, el uso compartido del mismo entre las aplicaciones, memoria virtual, I/O a dispositivos, etc. Esto se puede plasmar ya que los procesadores modernos soportan niveles de privilegios o rings. El sistema operativo, el supervisor, corre en el ring 0 (más privilegiado) y las aplicaciones en ring 3 (menos privilegiado). Para tal proceso las tecnologías de virtualización utilizan "ring deprivileging". Para explicar el concepto anterior observamos la figura posterior a este párrafo, donde se ilustra como el sistema operativo es modificado para poder ejecutarse en ring 1 dejando el ring 0 para el Hipervisor, de tal forma que tiene más poder que el sistema operativo para poder controlar los recursos a los cuales este puede acceder.

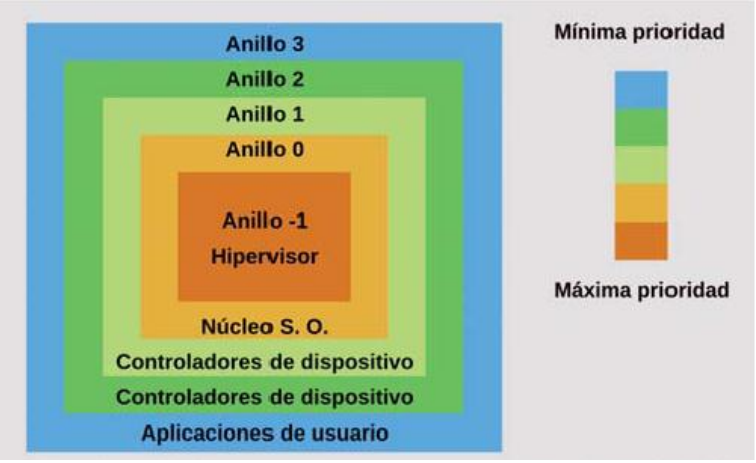


Figura 1. **Anillos de privilegio en arquitecturas x86 con soporte para virtualización. (Talens-Oliag, Sergio, 2009)**

1.5.2 Tipos de Hipervisores.

Los hipervisores se pueden clasificar en dos tipos:

- **Tipo 1: (nativo, baremetal o unhosted)**

Se trata de un software que se ejecuta directamente sobre el hardware real del equipo para controlarlo y monitorizar los sistemas operativos virtualizados, estos se ejecutan en otro nivel por encima del hipervisor.

Algunos de los hipervisores tipo 1 más populares son:

- VMware: ESXi (gratuito), ESX (de pago).
- Xen (libre).
- Citrix XenServer (gratuito).
- Microsoft Hyper-V Server (gratuito).

- **Tipo 2 (hosted)**

Este tipo de hipervisor se refiere a una aplicación que se ejecuta sobre un sistema operativo normal (Linux, Windows, MacOS) para virtualizar sistemas. Es así que esta forma de virtualización se da en una capa más alejada del hardware, lo que hace que el rendimiento sea menor en los hipervisores de este tipo.

Entre los Hipervisores tipo 2 más comunes son los siguientes:

- Oracle - Sun: VirtualBox (gratuito), VirtualBox OSE (libre).
- VMware: Workstation (de pago), Server (gratuito), Player (gratuito).
- KVM y QEMU (libres).
- Microsoft: Virtual PC, Virtual Server.

1.5.3 La librería libvirt

La librería libvirt es un API vinculada a las capacidades de virtualización de Linux, la misma que soporta una variedad de hipervisores entre los cuales resaltan Xen, KVM, QEMU y algunos otros productos de virtualización para otros sistemas operativos.

Tabla 1. Hipervisores que soporta libvirt. (Jones, 2009)

HIPERVISOR	DESCRIPCIÓN
Xen	Hipervisor para arquitecturas IA-32, IA-64, y Power PC. (IA - Intel y AMD)
Qemu	Emulador de plataformas para varias arquitecturas.
KVM Kernel-based Virtual Machine	Emulador de plataformas Linux.
LXC - Linux Container	Contenedor Linux (liviano) para virtualización del sistema operativo.
OpenVz	Virtualización a nivel de sistema operativo basado en el kernel de Linux.
Virtual Box	Hipervisor para virtualización de x86.
User Mode Linux (UML)	Emulador de plataformas Linux para varias arquitecturas.
Storage	Colección de unidades de almacenamiento (disco local, disco en red, volúmenes iSCSI.)

1.6 Plataformas de virtualización OpenSource

1.6.1 KVM (Kernel-based Virtual Machines)

"En Diciembre de 2006, Linus Torvalds anunció que las nuevas versiones del kernel Linux incluirían la herramienta de virtualización conocida como KVM (Kernel Virtual Machine Monitor), una tecnología de reciente aparición y su repentina aceptación se debe al poder del modelo de virtualización basado en el kernel. Esta ofrece ciertas ventajas potenciales, entre las que se encuentran un mejor rendimiento y un soporte más uniforme para el entorno Linux al completo.

En un escenario de virtualización típico, un componente conocido como Hipervisor ofrece una interfaz entre el sistema huésped y su anfitrión. El Hipervisor reside en lo alto del sistema anfitrión, encargándose de la planificación de las tareas y la gestión de la memoria de cada huésped. KVM integra el Hipervisor en el kernel, reduciendo así las redundancias y acelerando los tiempos de ejecución. Un controlador de KVM se comunica con el kernel actuando como interfaz para una máquina virtual en espacio de usuario. La programación de

las tareas y la gestión de la memoria son manejadas a través del mismo kernel. Un esquema o módulo del núcleo Linux presenta el modo huésped, instala tablas de páginas paraŽél y emula determinadas instrucciones clave.” (Shah, 2008)

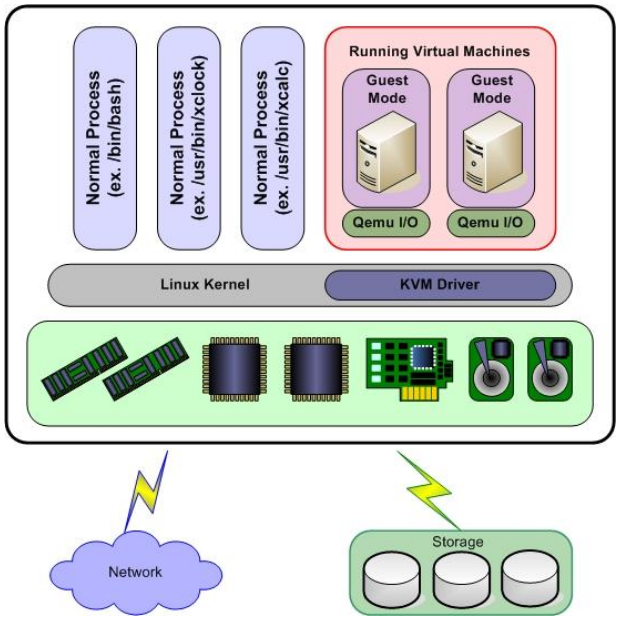


Figura 2. Arquitectura KVM. (IBM, 2009)

KVM (Kernel-Based Virtual Machines), se presenta como una solución de virtualización completa, cuyo proyecto fue iniciado por Qumranet, empresa que luego fue adquirida por Red Hat. Esta plataforma se ejecuta sobre GNU/Linux y depende de la extensión de virtualización de los procesadores como INTEL VT o AMD-V para poder ser implementada.

Todas las nuevas tecnológicas de desarrollo se están llevando ahora en relación a KVM, ya que en algunos años va a ser la solución más recomendada en el mundo Open Source debido a su integración al núcleo de Linux (a partir de la versión 2.6.18), por lo que se hereda algunas de las ventajas nativas de este sistema operativo; además de que está soportada por varios sistemas operativos como GNU/Linux, BDS, Solaris, MAC y Windows.

También hay mejoras en la seguridad entre las máquinas virtuales pero sobre todo brinda una configuración más amigable para los administradores.

Entre las características de KVM tenemos las siguientes:

- ✓ Está diseñado para procesadores x86, centrando en virtualización total.
- ✓ No se modifica el kernel de GNU/Linux.
- ✓ Solamente es un módulo que no necesita parches.
- ✓ Contiene soporte de paravirtualización.
- ✓ KVM funciona en todo tipo de máquinas, servidores, escritorio o laptop.
- ✓ Migración en caliente de máquinas virtuales.
- ✓ Tiene administración vía web, gráfica y consola.
- ✓ La administración de las máquinas virtuales se hacen por medio de un usuario normal.
- ✓ Podemos utilizar el comando kill para matar máquinas virtuales ya que en KVM son reconocidos como procesos.
- ✓ Manejo de redes virtuales.
- ✓ Permite ejecutar múltiples máquinas virtuales cada una con su propia instancia.

KVM puede funcionar en todo tipo de máquinas que tengan la extensión de procesador de Intel VT o de AMD Pacifica de cualquier tipo; es decir, se aplica tanto en servidores, equipos de escritorios y portátiles, pudiendo usarse las mismas herramientas de administración e infraestructura que usa Linux. El sistema KVM se integra con el planificador de Linux, la pila de E/S y todos los sistemas de archivos disponibles. Otros beneficios incluyen la migración durante la ejecución de la máquina virtual.

KVM (Kernel Virtual Machine Monitor) es una tecnología de virtualización que apareció recientemente en algunas distribuciones Linux y ahora es firmemente apoyado y soportado por distros como Ubuntu y la compañía *Red Hat* tras la adquisición de *Qumranet* en el 2008,

empresa que fue la precursora de KVM. El enfoque de esta herramienta está basado en un *Hipervisor* y su funcionamiento está íntimamente ligado en el kernel, lo que ofrece ciertas ventajas potenciales, entre las que se encuentran un mejor rendimiento y un soporte más uniforme para el entorno Linux al completo.

1.6.2 XEN

"Xen fue inicialmente un proyecto de investigación de la Universidad de Cambridge (la primera versión del software fue publicada a fines de 2003). Este proyecto fue liderado por Ian Pratt, quien luego formó una empresa -junto con otras personas- para dar servicios de valor agregado como soporte, mantenimiento y capacitación sobre Xen en Enero de 2005.
Dado que Xen está licenciado bajo GPL[9] el código no puede cerrarse, y no es solo Xensource quien mantiene el código, sino que varias empresas importantes como IBM, Sun, HP, Intel, AMD, RedHat, Novell están sumamente involucradas en el desarrollo asignando programadores al mantenimiento de este software." (Woitasen, 2006)
A pesar de que la empresa XenSource desde el 2007 es propiedad de la empresa Citrix, hay que aclarar que el hipervisor Xen como tal, no es comercial por su carácter de GPL, sino tan solo lo son las herramientas de administración que desarrolla Citrix. Es así que la comunidad abierta Xen.org es la que desarrolla y mantiene el Hipervisor Xen como una solución gratuita bajo la licencia GNU General Public License.

Xen es un Hipervisor de máquinas virtuales (Open Source), el mismo que permite ejecutar varias instancias de sistemas operativos, con todas sus características, de forma completamente funcional sobre un mismo equipo físico. Entre varias de sus características proporciona aislamiento seguro, control de recursos, garantías de calidad de servicio y migración de máquinas virtuales en vivo. A diferencia de otras tecnologías de máquinas virtuales, requiere portar los sistemas operativos para adaptarse al API de Xen (aunque manteniendo la compatibilidad con aplicaciones de usuario).

[9]GPL: General Public License (Licencia Pública General). Licencia creada por la Free Software Foundation y orientada principalmente a los términos de distribución, modificación y uso de software libre.

Empezaremos con algunos conceptos y características de los componentes importantes que forman parte de esta arquitectura hasta terminar en algunos detalles que deben ser tomados en cuenta antes de la configuración e implementación.

> *"Xen es un software que permite implementar máquinas virtuales. Este concepto se refiere a tomar los recursos de una máquina física y compartirlos entre distintas instancias de uno o varios sistemas operativos ejecutándose en forma concurrente. Desde el punto de vista del usuario final, cada una de esta instancias tendrá las mismas prestaciones que un sistema operativo común corriendo sobre una computadora tradicional, siendo la virtualización algo totalmente transparente".* (Clark, 2008)

Xen a diferencia de otras tecnologías de virtualización que corren sobre un sistema operativo, se ejecuta directamente sobre el hardware. No es un componente adicional del Kernel de Linux ni una aplicación, sino que es un programa completamente independiente lo que beneficia al rendimiento al evitar la traducción de instrucciones y a la fiabilidad ya que está aislado de problemas, bugs o inestabilidad del kernel. Cuando inicia un sistema virtualizado con Xen, es éste quien bootea y luego inicia las máquinas virtuales.

Los dispositivos principales que se virtualizan en Xen son el disco rígido y la placa de red. Además de los dispositivos anteriores existen otras características muy útiles y que están en continuo desarrollo por parte de la comunidad (xen.org) y que listamos a continuación:

- ✓ Velocidad (excelente rendimiento de entre 2% a 8% de penalización en carga de procesador).
- ✓ Código fuente simple y reducido (menos de 50.000 líneas de código).
- ✓ Excepcional particionamiento de recursos de E/S de bloques y red.
- ✓ Optimización de CPU y memoria, con soporte hardware Intel VT y AMD Pacífica.

- ✓ La posibilidad de "migrar en caliente" máquinas virtuales de un equipo de hardware a otro.
- ✓ Soporte para servidores virtuales con sistema operativo multiprocesador de 32 vías o bits.
- ✓ Soporte para la virtualización por hardware de los Intel VT-X y los AMD Pacifica, que permite ejecutar sistemas operativos sin modificar dentro de las máquinas virtuales (Windows XP/2003 y UNIX antiguos).
- ✓ Soporte para Intel PAE (Physical Addressing Extensions) en servidores de 32 bits con más de 4 GB de RAM.
- ✓ Soporte para las arquitecturas x86/64 (tanto AMD64 como EM64T).
- ✓ Soporte para SMP (Symmetric Multi-Processor).
- ✓ Herramientas de control.
- ✓ Soporte de ACPI (Advanced Configuration and Power Interface).
- ✓ Soporte para gráficos AGP/DRM[10] (gráficos 3D).

Esta tecnología de virtualización también tiene ventajas que la hacen destacarse de otras alternativas de virtualización debido a que:

Primeramente es Open Source, lo que deriva una mejor funcionalidad, mejor rendimiento, gran extensibilidad y seguridad, ya que está desarrollado por una comunidad de muchos miembros en todo el mundo. Actualmente se considera a este hipervisor como el de mejor rendimiento, producto de su tecnología de paravirtualización, la cual permite la colaboración de los servidores hospedados para conseguir el mejor rendimiento en aplicaciones corporativas.

Actualmente Xen puede virtualizar completamente sistemas operativos propietarios como Windows, sin límite de servidores y se puede hacer migraciones de hardware en caliente, en

[10]AGP: siglas de Puerto acelerador de gráficos (Accelerated Graphics Port) y DRM son las siglas de Gestor de renderizado directo (Direct Rendering Manager).

un servidor XEN pueden convivir servidores paravirtualizados y completamente virtualizados con la ventaja adicional de que en distribuciones como Red Hat 5 y CentOs, Xen viene integrado más las opciones de clusterización, las mismas que son muy útiles en entornos de alta disponibilidad (HA).

En cuanto a la virtualización completa, Xen usa de manera óptima las capacidades de virtualización por hardware de los procesadores VT de Intel y los Pacifica de AMD, ya que trabaja en conjunto con sus fabricantes.

1.6.3 OpenVZ

OpenVZ brinda virtualización a nivel del kernel y del sistema operativo de código abierto lo que ofrece una menor flexibilidad ya que tanto los huéspedes como los anfitriones deben ser Linux. Sin embargo, la virtualización a nivel de sistema operativo en OpenVZ proporciona mejor rendimiento, escalabilidad, densidad, administración de recursos dinámicos, y facilidad de administración que otras alternativas.

En la web hemos captado muchas comentarios buenos de esta tecnología de virtualización, pero el hecho de que Open VZ no trabaje con sistemas operativos que no son Open Source, lo cataloga como una opción que no consideraremos por el momento ya que por las características del proyecto nuestra plataforma a escoger debe soportar la mayoría de sistemas tanto libres como propietarios.

1.6.4 VirtualBox

Virtualbox es otra plataforma de virtualización bajo el enfoque de emulación de sistema operativo para sistemas de 32 bits tanto Windows, Linux (aunque para Linux es necesario instalar una serie de librerías adicionales), y para entornos MAC se está trabajando en su desarrollo.

Ahora virtual Box es propiedad de Oracle sin embargo existe la versión OSE (Open Source Edition) y se indica que la nueva aplicación dará el salto a nuevas versiones con interfaz de usuario mejorada, además de soporte adicional para hardware virtual, incluyendo los chipsets que dan soporte a tarjetas de expansión PCI Express.

Los nuevos aditamentos implican un aumento de la capacidad soportada mediante el uso de un nuevo modelo asíncrono de entrada salida para almacenamiento en red y local, además de un mejor soporte de los estándares del llamado Open Virtualization Format (OVF).

Debido a que usa tecnología basada en emulación su rendimiento es menor que las basadas en paravirtualización por lo que su uso es recomendado para entornos de escritorio y no en entornos de producción.

A continuación se presenta un extracto de un cuadro comparativo de las más conocidas soluciones en cuanto a virtualización Open Source se refiere:

Tabla 2. Cuadro comparativo de las distintas plataformas OpenSource de virtualización.

	Virtualizacion Completa	Paravirtualización	Virtualización a nivel de S.O.	Licencia	Arquitecturas soportadas	Rendimiento	Soporte para invitados SMP (multiprocesamien	Conexion en caliente de CPU/Memoria	Autonomía	NOTAS
XEN	x	x		GPL	i686, x86_64, IA64, PPC	Paravirtualizacion - super rápido, Virtualización completa - medio	x	x	x	Solo virt. completa necesita extensiones Intel VT y Pacifica AMD
KVM	x	x		GPL	i686, x86_64	Paravirtualizacion - super rápido, Virtualización completa - medio				Paravirt. Y Virt. Completa necesita extensiones Intel VT y
VIRTUAL BOX	x			GPL/ PROPIETARIO	i686, x86_64	Virtualización Completa - Rápido / Super rápido				Modulo Kernel GPL, el modulo UBS es propietario.
OPEN VZ			x	GPL	TODOS LOS LINUX	Nativo				Migración en caliente.
VSERVER			x	GPL	TODOS LOS LINUX	Nativo				Mal rendimiento en aislamiento.

De las opciones analizadas se ha tomado especial consideración a XEN, por ser actualmente la más desarrollada y usada en entornos de virtualización Open Source. Tiene largo tiempo de desarrollo (a partir del 2007).Se diferencia en que tiene una arquitectura en donde para usar el concepto de paravirtualización, no se necesita tener soporte (o extensiones) a nivel de procesador salvo que se necesite virtualización completa. A diferencia de las otras opciones que, tanto para máquinas paravirtualizadas como de

virtualización completa, necesitan contar con procesadores que tengan soporte o extensiones de virtualización. Sin embargo, mencionamos brevemente un aspecto relevante en cuanto a la plataforma basada en KVM, no es muy madura aún pero está ganando respaldo de algunas distribuciones importantes de Linux como son Ubuntu y recientemente Red Hat; lo que la convierte en una tecnología muy prometedora a futuro. En cuanto al rendimiento de KVM no se compara con XEN, debido a que se clasifica como de tipo hipervisor tipo 2; por tanto no ofrece el mismo rendimiento que trabajar bajo un hipervisor de tipo 1 como Xen. Sumado a esto, no poseen la propiedad de aislamiento como la tiene Xen en la que el fallo de una máquina virtual no implica ninguna interferencia sobre el resto de máquinas. Además podemos aseverar que al tener Xen más tiempo de desarrollo, ha sido testeada de mejor forma; por tanto existe mucho más soporte para distintas arquitecturas y los problemas (o bugs) derivados de cada una de estas.

Para el caso específico de la UTPL conforme a los objetivos planteados en este proyecto de investigación, se ha tomado en cuenta solo este tipo de plataformas Open Source, de las cuales aplica Xen como candidata por su amplia aceptación en el campo de la virtualización, por el concepto de paravirtualización y algunas ventajas abordadas anteriormente. Además de tener un buen rendimiento en virtualización completa, se destaca por su capacidad de soportar la mayoría de arquitecturas de procesador y sistemas operativos, sus amplias funcionalidades y sobre todo por el alto grado de madurez en desarrollo que es importante en el mundo del desarrollo de software y en consecuencia de la Virtualización Open Source. Es así que tenemos algunas ventajas extra sobre las otras, entre las que mencionamos:

- ✓ *Puede trabajar sobre equipos con o sin soporte para virtualización completa esto es sin la extensión a nivel de procesador (Intel-VT y Pacífica de AMD).*
- ✓ *Por la capacidad de trabajar con dominios o equipos virtuales modificados o paravirtualizados.*

- ✓ *Por la posibilidad de configurarse sobre entornos de clustering y de alta disponibilidad (HA), mitigando una de las mayores debilidades de la virtualización que es el de tener un punto central de fallo.*
- ✓ *Esta plataforma tiene un mejor acoplamiento para lograr simplificar la infraestructura de equipos y de consumo de energía acorde al entorno y capacidad de nuestros servidores de la Universidad y por ende para este proyecto.*

CAPÍTULO II

ANÁLISIS DE CRITICIDAD DE SERVICIOS

2.1 Metodología

La metodología de clasificación y evaluación se basará en calificación de variables definidas, en la que se tomarán los resultados arrojados por una encuesta aplicada a cada uno de los administradores de servicios y/o servidores que laboran en la UTPL, en donde se califica cada uno de estos por el nivel de severidad - riesgo. En este punto de selección se separarán algunos, tomando en cuenta parámetros como servicios ya virtualizados, otros que interactúan con dispositivos especiales como sensores biométricos, bases de datos distribuidas u otros con alto consumo de recursos y otros parámetros que se mencionan más adelante en la fase de selección.

2.2 Evaluación de la criticidad

La criticidad es una medida de ponderación que considera los siguientes puntos:

- El **efecto (severidad de riesgo)** en caso de falla de un componente funcional (o equipo) dentro de un determinado proceso.
- La **dependencia de funcionalidad** de un servicio específico con otras aplicaciones con que se interactúe.
- La **velocidad** de reparación de la falla y continuidad del servicio.
- La **frecuencia (caídas del servicio):** es el intervalo de ocurrencia de la falla o caída de un servicio específico.

En esta evaluación lo fundamental es tomar la criticidad como un indicador que nos permita ver de mejor manera, la magnitud del problema ocasionado por la falla de un servicio(s) y/o equipo y sus repercusiones en las actividades operacionales de la Universidad. Una vez obtenido este nivel de criticidad, éste será empleado para definir una clasificación que nos brinde una perspectiva clara de importancia de cada uno de estos servicios.

2.3 Servicios informáticos existentes en la UTPL

Las actividades de la Universidad requieren de la operación de una variedad de servicios, los mismos que permiten que las actividades se desarrollen con eficacia aprovechando la infraestructura existente. Entre los más conocidos están:

Tabla 3. Lista de servicios UTPL.

1	**AQCT**	PROGRAMACIÓN CENTRAL, TARIFACION, PROGRAMACIÓN IP SIEMMENS.
2	**ASTERISK BACKUP**	SERVIDOR DE BACKUP ASTERISK Y PRUEBAS ELASTIX
3	**ASTERISK VoIP**	SERVIDOR DE VOZ SOBRE IP
4	**BACKUP LDAP**	SERVIDOR BACKUP DE AUTENTICACIÓN LDAP
5	**BACKUP SERVIDOR WEB**	SERVIDOR DE RESPALDO DEL SERVIDOR WEB Y SERVICIO OCW
6	**BASE DE DATOS - CAJANUMA**	SERVIDOR DE BASE DE DATOS DE LOS APLICATIVOS DE LA INSTITUCION
7	**BASE DE DATOS - PALTAS**	BASE DE DATOS PARA DESARROLLO DE APLICACIONES
8	**BD-RRHH**	BASE DE DATOS DEL DEP. DE RECURSOS HUMANOS
9	**CENTRAL IP SIEMMENS**	COMUNICACIÓN ENTRE CENTRALES QUITO - LOJA
10	**CENTRAL TELEFÓNICA**	COMUNICACIÓN INTERNA Y EXTERNA
11	**CTMAIL**	TRANSFERENCIA DE LLAMADAS EXTERNAS.
12	**DGCITTES**	SERVIDOR QUE SE ENCUENTRA EN EL DSPACE - REPOSITORIO PUBLICACIONES
13	**DNS CACHING**	SERVIDOR CACHING DE RESOLUCION DE NOMBRES DOMINIO
14	**DNS EXTERNO**	SERVIDOR DE DOMINIOS EXTERNO
15	**DNS INTERNO**	SERVIDOR DE RESOLUCION DE NOMBRES INTERNO
16	**EVA**	SERVIDOR DEL ENTORNO VIRTUAL DE APRENDIZAJE
17	**F-SECURE ANTIVIRUS**	ANTIVIRUS CORPORATIVO UTPL
18	**IME**	SERVIDOR DE LOGS DE IPS
19	**INGLES**	SISTEMA DE EXÁMENES DE INGLES MULTIMEDIA
20	**IQA**	CLUSTER DE ALTO RENDIMIENTO PARA EL IQA
21	**IVR**	RESPUESTA AUTOMÁTICA DE INFORMACIÓN
22	**LDAP**	SERVIDOR PRINCIPAL DE AUTENTICACIÓN LDAP
24	**MAIL (GDR3)**	SERVIDOR DE MAIL
25	**NOC**	SERVICIO DE MONITOREO ACTIVO DE EQUIPOS, SERVICIOS, RECURSOS Y REDES.
26	**NOCAT**	PORTAL CAUTIVO - RADIUS (AUTENTICACION DE EQUIPOS)
27	**OSSIM**	MONITOREO DE SEGURIDAD
28	**PRESENTATION SERVER**	ADMINISTRADOR DE SESIONES (VIDEOCONFERENCIA)
29	**PROXY 2**	SERVIDOR PROXY SECUNDARIO
30	**PROXY -DHCP**	SERVIDOR PROXY - ACCESIBILIDAD INTERNET - DHCP
31	**REPOSITORIO**	SERVIDOR DE REPOSITORIO PARA COMUNIDAD OPENSOURCE
32	**SAS**	COMPARTIR APLICACIONES (SHARED)
33	**SERVIDOR WEB**	SERVIDOR WEB DE LA UTPL.
34	**TIVOLI**	GESTION ACTIVOS - MESA DE SERVICIOS
35	**TSTULEE**	SERVIDOR DE PRUEBAS DEL LABORATORIO VIRTUAL DE INGENIERIA SISMICA
36	**VIRTUALIZACION I**	SERVIDOR PARA VIRTUALIZACION CON VMWARE ESX
37	**VIRTUALIZACION II**	SERVIDOR 2 PARA VIRTUALIZACION CON VMWARE ESX

38	**VLEE**	SERVIDOR DE PRODUCCION DEL LABORATORIO VIRTUAL DE INGENIERIA SISMICA
39	**VPN**	ACCESO A SERVICIOS DE INTRANET
40	**AUDITORIA SEGURIDAD**	AUDITORIA DE SEGURIDAD DE REDES - BACTRACK 4
41	**OWASP**	AUDITORIA DE APLICACIONES WEB ABIERTAS.

Entre algunos de estos servicios ya existen soluciones de virtualización propietarias adquiridas por nuestra Universidad (VMWARE ESX), que están en funcionamiento y que por tanto se descartarán de la selección de servicios.

2.4 Elaboración de la matriz de riesgo.

Para evaluar la efectividad de la gestión y administración de los riesgos que impactan sobre los objetivos de un determinado proyecto se necesita de una ***matriz de riesgo***. La matriz de riesgo es una herramienta de control y de gestión generalmente usada para poder identificar las actividades más importantes dentro de la planificación y elaboración de un proyecto o actividad. Determinan el tipo y nivel de riesgos asociados a estas actividades y los factores de riesgo con los efectos que se ahí se generan. En nuestro caso esta matriz estará relacionada con identificar y cuantificar variables que establezcan un nivel de criticidad de los servicios existentes en la UTPL.

a) **Matriz de Riesgo:** este tipo de matriz permitirá asignar un valor de riesgo a un servicio, en virtud de la aplicación de criterios previamente definidos.

b) **Levantamiento de la criticidad:**

En esta parte se realizó una escala para medir el nivel de contribución que presentan los distintos servicios al cumplimiento de actividades y procesos en la Universidad:

Tabla 4. Tabla de niveles de riesgo.

Clasificación de Nivel	Descripción de nivel de contribución	Valor
Alto	El servicio influye de manera fundamental en el cumplimiento de actividades y procesos estratégico de la UTPL.	3
Medio	El servicio influye de manera importante en el cumplimiento de actividades y procesos de la UTPL.	2
Bajo	El servicio influye en menor proporción en el cumplimiento de las actividades en la UTPL.	1
Nulo	No aporta en el cumplimiento de actividades y procesos en la UTPL.	0

Al estar íntimamente relacionados los riesgos con la determinación de criticidad es necesario realizar una matriz donde se analizan las distintas variables que permitirán evaluar de manera más precisa la criticidad de cada uno de los servicios analizados.

MATRIZ DE RIESGO

•Escala para la probabilidad de ocurrencia:

Tabla 5. Niveles de probabilidad de riesgo.

Categoría	Valor	Descripción
Casi certeza	5	Riesgo cuya probabilidad de ocurrencia es muy alta, es decir se tiene plena certeza que se presente, tiende al 100%.
Probable	4	Riesgo cuya probabilidad de ocurrencia es alta, es decir se tiene entre 75% a 95% de seguridad que se presente.
Moderado	3	Riesgo cuya probabilidad de ocurrencia es media, es decir, se tiene entre 51% a 74% de seguridad que se presente.
Improbable	2	Riesgo cuya probabilidad de ocurrencia es baja, es decir, se tiene entre 26% a 50% de seguridad que se presente.
Muy improbable	1	Riesgo cuya probabilidad de ocurrencia es muy baja, es decir se tiene entre 1% a 25% de seguridad que se presente.

Escala para el Impacto:

Tabla 6. Niveles de impacto.

Categoría	Valor	Descripción
Catastróficas	5	Riesgo cuya materialización influye directamente en las actividades de la Universidad, dejando inoperativo el servicio por un periodo importante de tiempo.
Mayores	4	Riesgo cuya materialización dañaría significativamente el servicio y por tanto las actividades de la Universidad. Además se requeriría una cantidad importante de tiempo de los administradores en investigar las causas y reparar los daños.
Moderadas	3	Riesgo cuya materialización causaría daño y pérdida de tiempo a los administradores, pero sin mayor efecto sobre las actividades regulares en la Universidad.
Menores	2	Riesgo que causa un daño en el servicio que se puede corregir en el corto tiempo y que no afecta a las actividades de la institución.
Insignificantes	1	Riesgo que puede tener un efecto nulo sobre la institución.

Escala para medir el nivel de severidad del Riesgo:

Tabla 7. Nivel de severidad de riesgo.

Nivel Probabilidad (P)	Nivel Impacto (I)	Nivel Severidad (P * I)
Casi certeza (5)	Catastróficas (5)	EXTREMO (25)
Casi certeza (5)	Mayores (4)	EXTREMO (20)
Casi certeza (5)	Moderadas (3)	EXTREMO (15)
Casi certeza (5)	Menores (2)	ALTO (10)
Casi certeza (5)	Insignificantes (1)	ALTO (5)
Probable (4)	Catastróficas (5)	EXTREMO (20)
Probable (4)	Mayores (4)	EXTREMO (16)
Probable (4)	Moderadas (3)	ALTO (12)
Probable (4)	Menores (2)	ALTO (8)
Probable (4)	Insignificantes (1)	MODERADO (4)
Moderado (3)	Catastróficas (5)	EXTREMO (15)
Moderado (3)	Mayores (4)	EXTREMO (12)
Moderado (3)	Moderadas (3)	ALTO (9)
Moderado (3)	Menores (2)	MODERADO (6)
Moderado (3)	Insignificantes (1)	BAJO (3)
Improbable (2)	Catastróficas (5)	EXTREMO (10)
Improbable (2)	Mayores (4)	ALTO (8)
Improbable (2)	Moderadas (3)	MODERADO (6)
Improbable (2)	Menores (2)	BAJO (4)
Improbable (2)	Insignificantes (1)	BAJO (2)
Muy improbable (1)	Catastróficas (5)	ALTO (5)
Muy improbable (1)	Mayores (4)	ALTO (4)

La severidad del riesgo la calculamos realizando un producto de las variables de probabilidad y el impacto (P x I). Los datos de probabilidad e impacto fueron suministrados por los administradores de cada uno de estos servicios y/o servidores a partir de la aplicación de encuestas individuales. De aquí se seleccionarán aquellos servicios que por su criticidad baja son tomados en cuenta para establecer si son factibles de ser virtualizados o no.

2.5 Selección de los servicios menos críticos de la UTPL.

En el análisis realizado a la UTPL tanto en su infraestructura de servidores y en sus servicios, se ha usado adicionalmente algunas variables o parámetros que influirán en el establecimiento de una valoración clara sobre cada uno de los servicios. Además de lo anterior se han incluido resultados de encuestas, aplicadas al noventa por ciento de los administradores de cada uno de estos servicios y servidores, cuyo aporte se ha unido a

estadísticas de rendimiento obtenidas en reportes de la administración de servidores (anexo), lo que ha derivado en una categorización bajo las siguientes variables:

1) Efectos ante una eventual caída de un servicio específico (severidad de riesgo).

Tabla 8. Caídas de servicio.

Severidad de Riesgo	Cantidad
EXTREMO	18
ALTO	8
MODERADO	8
BAJO	7
Total general	**41**

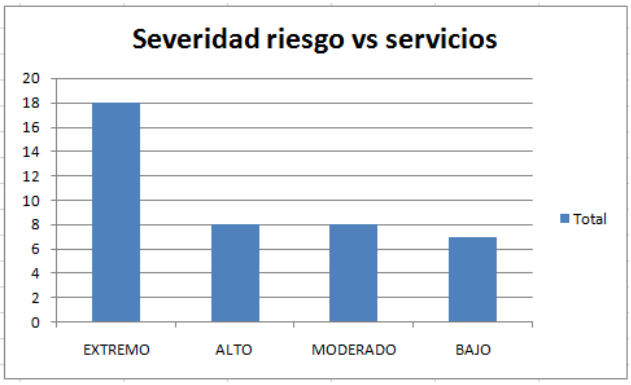


Figura 3. Severidad riesgo vs servicios.

2) Número de usuarios que interactúa con cada servicio.

Tabla 9. Número de usuarios por servicio.

Num. Usuarios	Total servicios
DE 1 A 100	23
MAYOR A 1000	11
DE 100 A 500	5
DE 500 A 1000	2
Total general	**41**

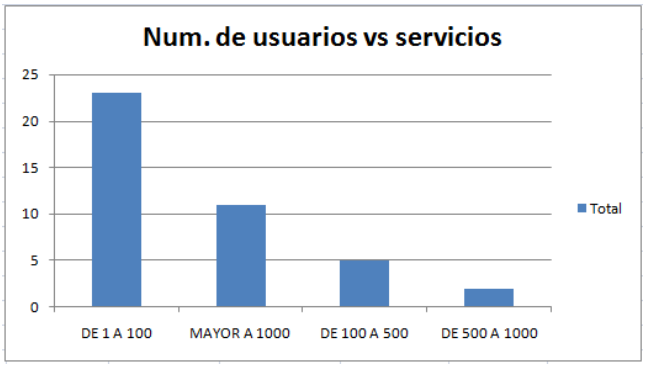


Figura 4. Número de usuarios vs servicios.

En esta parte se tomó en cuenta los servicios con el número mínimo de usuarios, por lo tanto es un indicador de bajo nivel de influencia en la criticidad de un servicio.

3) Rendimiento de procesador, memoria y almacenamiento de cada servidor.

Tabla 10. Datos servidores DELL. (UTPL - Picoita, Galo, 2009)

ITEM	Servidor	Administrador	Dependencia	Modelo Server	Tipo Procesador	Procesador			Memoria		Disco Interno			Fecha de Producción	Proyección en años
						Num.	Velocidad GHZ	Uso %	Tamaño GB	Uso %	Num.	Capacidad GB	Uso %		
DELL1	BDDGDS	Viviana Montaño	Grupo de Desarrollo de Software	DELL-PowerEdge 1600SC	Intel(R) Xeon(TM)	2	2.8	6%	3.93	45%	2	60.0 GB 76.4 GB	80%	01/10/2004	5
DELL2	DEVCRM	Viviana Montaño	Grupo de Desarrollo de Software	DELL-PowerEdge 1600SC	Intel(R) Xeon(TM)	2	2.8	2%	3.93	60%	1	60 GB	75%	01/09/2004	5
DELL3	DEVSERVER.OLD	Viviana Montaño	Grupo de Desarrollo de Software	DELL-PowerEdge 1600SC	Intel(R) Xeon(TM)	2	2.8	50%	3.93	80%	1	60 GB	98%	01/09/2004	5
DELL4	TSTSERVER	Viviana Montaño	Grupo de Desarrollo de Software	DELL-PowerEdge 1600SC	Intel(R) Xeon(TM)	2	2.8	25%	2.5	55%	1	60 GB	98%	02/09/2004	5
DELL5	PRESENTATION SERVER	Rodrigo Barba	Unidad de Video Conferencias	DELL-PowerEdge 2600SC	INTEL XENON TM CPU 2.8 GHZ	1	2.8	50%	1	80%	1	30GB	60%	08/01/2004	10
DELL6	PRODUCCION	Rodrigo Barba	Unidad de Video Conferencias	DELL-PowerEdge 1600SC	Intel(R) Xeon(TM)	1	2.8	40%	1.5	40%	1	80GB	25%	10/01/2007	5
DELL7	VC GRABADOR	Rodrigo Barba	Unidad de Video Conferencias	DELL-PowerEdge 1600SC	Intel(R) Xeon(TM)	1	2.4	20%	1	30%	1	160	35%	02/01/2009	5
DELL8	VIDEOCONTV	Rodrigo Barba	Unidad de Video Conferencias	DELL-PowerEdge 1600SC	Intel(R) Xeon(TM)	1	2.4	50%	1	20%	1	160	20%	10/01/2008	5

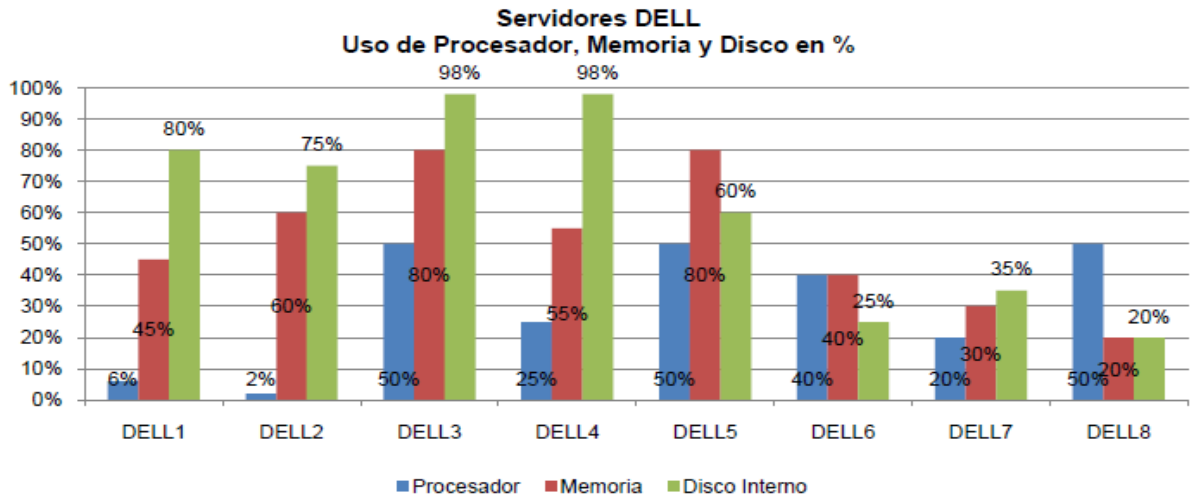


Figura 5. Estadísticas servidores DELL.

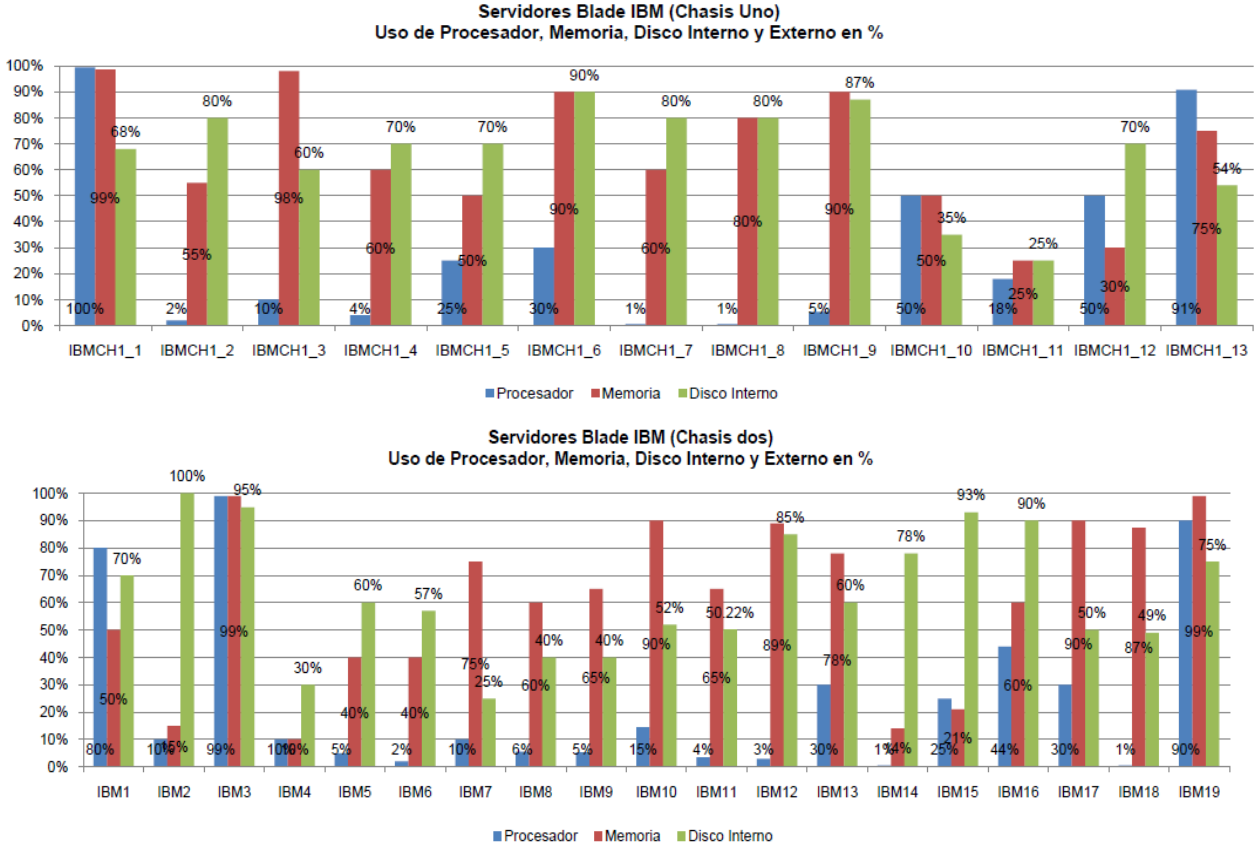


Figura 6. Estadísticas servidores IBM. **(UTPL - Picoita, Galo, 2009)**

*NOTA: Para más detalles de la nomenclatura de cada servidor IBM, en el **Anexo B** se puede revisar los detalles correspondientes a cada servidor.*

De los equipos sujetos a monitoreo, se obtuvo datos estadísticos de rendimiento pertenecen a la sala de servidores y otras dependencias de los cuales tomamos un dato que nos interesa que es el porcentaje de utilización de procesador (en situaciones normales), aquí tenemos que el promedio es de un 48.9%; de los cuales encontró que 22 de los 39 equipos tienen una utilización por debajo del 20% de la capacidad total del procesamiento de cada servidor lo que claramente implica una subutilización de estos equipos.

4) Entorno de funcionamiento(Producción, desarrollo y test)

Tabla 11. Entorno de funcionamiento.

Entorno de Operación	Cantidad
Producción	55
Pruebas	10
Desarrollo	3
Total general	**68**

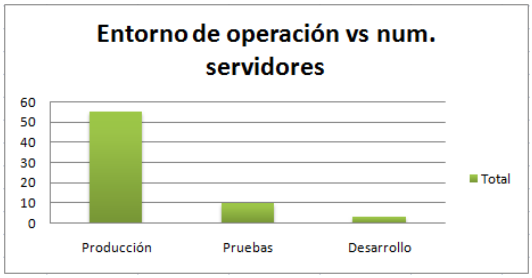


Figura 7. Entorno de operación vs numero de servidores.

Aquí se dará prioridad a servicios que estén en proceso de desarrollo o test, porque estos son ideales para una implementación de virtualización por su condición de cambios frecuentes en configuraciones.

5) Ubicación física de funcionamiento del servidor o servicio.

Tabla 12. Servidores por dependencias.

Dependencias	Numero de servidores
Grupo de Desarrollo de Software	23
Grupo de Telecomunicaciones	22
Soporte Técnico	4

Unidad de Video Conferencias	4
Unidad de Virtualización	3
Unidad Civil Geologia y Minas	2
Gestión del Conocimiento	2
Hospital UTPL	2
Call Center	1
Cursos Especializados	1
DG. Recursos Humanos	1
Dir. G CITTES	1
RESEC	1
Sistemas de Información Geográfica	1
Total general	**68**

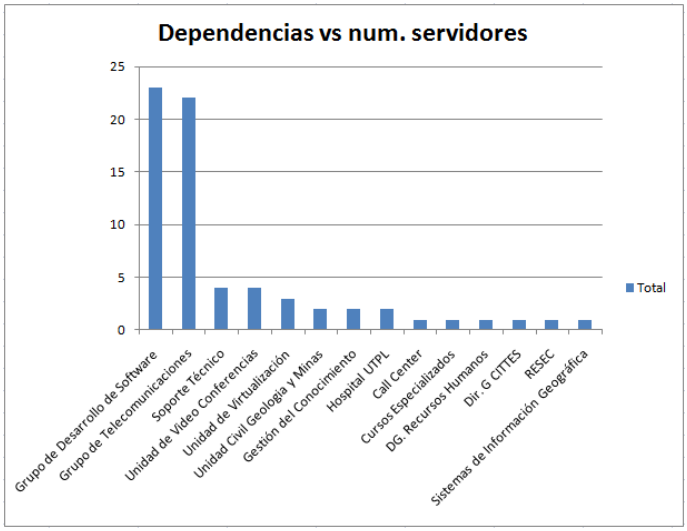


Figura 8. Servidores por dependencia.

De todas las dependencias listadas ubicamos aquellos servicios que estén dentro del campus de la Universidad e identificaremos a las dependencias o unidades productivas con un alto porcentaje en demanda de servidores, especialmente a los que tengan ubicación física en la sala de servidores que es donde se concentra físicamente la mayor parte de toda la infraestructura de servidores.

❖ *Sistemas Operativos usados en los servidores.*

Tabla 13. Sistemas operativos usados en servidores.

S.O.	No de S.O. por servidor
Windows 2003 Server	25
Centos	17
Solaris	7
Ubuntu	4
Windows XP	4
Debian	2
Slackware 12	2
UNIX AIX	2
Windows Server 2008	2
Mac os X	1
Seleccione SO * (sin especificar)	2
Total general	**68**

Figura 9. Sistemas operativos usados en servidores.

Estos datos nos muestran que existen sistemas operativos en su mayoría, soportados por Xen a excepción de Mac OsX y Unix AIX, ya que estas plataformas en la actualidad aún no tienen soporte para ser virtualizados.

- **Caídas del servicio por año vs servicios /servidores.**

Tabla 14. Caídas de servicio por año.

Num. caídas por año	cantidad servidores
DE 1 A 5	30
MAYOR A 10	4
DE 5 A 10	3
NINGUNA	1
Total general	**38**

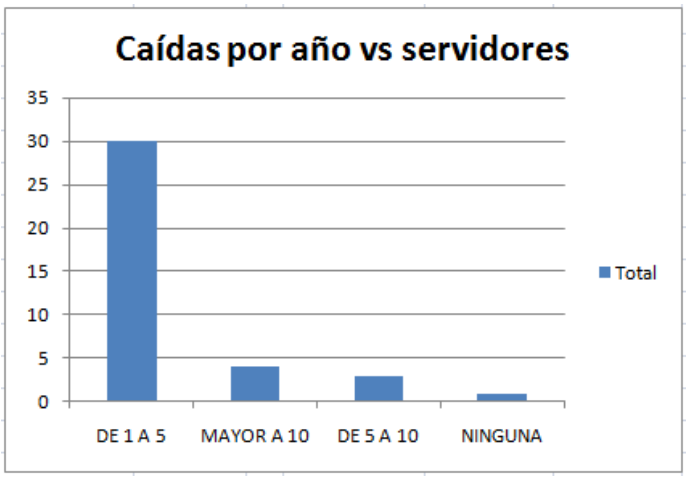


Figura 10. Caídas de servicio por año.

✓ Luego de clasificar y seleccionar aplicando las variables de criticidad antes mencionadas, tenemos como resultado el siguiente cuadro:

Tabla 15. Servicios seleccionados para ser virtualizados.

										PROBABILIDAD		IMPACTO					
SERVICIOS	DESCRIPCIÓN SERVICIO(s)	ENTORNO	NUM. USUARIOS	INTERACCIÓN CON OTROS SISTEMAS	CAIDAS SERVICIO POR AÑO	PROCESADOR	RAM	TIPO SERVER	S.O.	CLASIFICACION	VALOR	CLASIFICACION	VALOR	SEVERIDAD DEL RIESGO	VALOR	EFECTOS CAIDA SERVICIO PARA UTPL	OBSERVACIONES
DNS CACHING	SERVIDOR CACHING DE RESOLUCION DE NOMBRES DOMINIO	PRODUCCION	DE 1 A 100	NO	DE 1 A 5	INTEL PENTIUM II	3 GB	FISICO	CENTOS 5	MUY IMPROBABLE	1	MENOR	2	BAJO	2	NOS QUEDARIAMOS SIN DNS CACHING Y FUNCIONARIA	DNS PRINCIPAL ASUMIRIA TODA LA CARGA DE CONSULTAS
SERVIDOR DE AUDITORÍA	SERVIDOR DE AUDITORÍA DE SEGURIDAD DE LA	PRODUCCION	DE 1 A 100	NO	NINGUNA	INTEL XEON	3 GB	FISICO	BACKTRACK 4	IMPROBABLE	2	MENOR	2	BAJO	4	NINGUNO	SERVICIO OCASIONAL
OSWASP	SERVIDOR DE AUDITORIA DE APLICACIONES WEB ABIERTAS.	PRODUCCION	DE 1 A 100	NO	NINGUNA	INTEL XEON	3 GB	FISICO	SLAX WEBGOAT Y LABRAT	IMPROBABLE	1	MENOR	3	BAJO	3	MINIMO. LAS APLICACIONES WEB NUEVAS NO PODRÍAN SER TESTEADAS EN BUSCA DE FALENCIAS.	SERVICIO OCASIONAL

El cuadro con los parámetros de análisis completo de esta clasificación lo podemos encontrar en el *[ANEXO A].*

Ocho servicios han sido catalogados como no críticos para la UTPL, de los cuales tres han sido escogidos por su bajo riesgo de impacto en las actividades de la UTPL:

- Servicio de cache resolución de nombres de dominio cache.

  La influencia en las actividades y operatividad de la UTPL es mínima. Por su bajo consumo de recursos en hardware, sumado a que existen servidores principales los mismos que asumirían la totalidad de carga de consultas de resolución de dominios.
- Servicio de Auditoría de aplicaciones web abiertas OWASP y
- Servicio de Auditoría de redes con su distribución Bactrack versión 4.

  Estos dos servicios también aplicaron por su bajo número de usuarios, bajo consumo en recursos hardware y por su uso ocasional al realizar las pruebas de auditoría de redes y aplicaciones web respectivamente.

En está sub-clasificación de servicios menos críticos se realizó una evaluación de factibilidad, tomando en cuenta parámetros y limitaciones descritas en observaciones brindadas por cada administrador del servicio y la capacidad de los equipos en los que serán implementados los servicios finalmente seleccionados.

CAPITULO III

ARQUITECTURA DE LA PLATAFORMA DE VIRTUALIZACIÓN XEN.

3.1 Arquitectura XEN.

3.1.1 Elementos de Xen

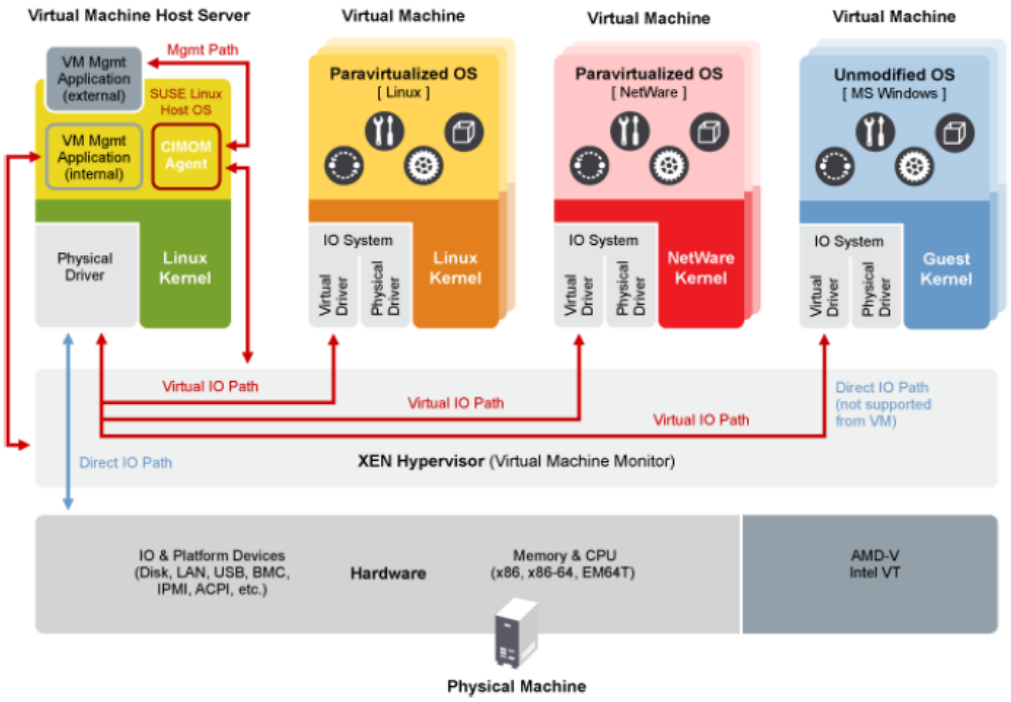


Figura 11. Arquitectura de XEN. (Vavai, 2010)

En la figura anterior se visualizan todos los componentes que forman parte de esta arquitectura en donde se destaca el hardware físico real, el Dom0 u Host, las máquinas virtuales y sobre todo la capa del hipervisor o Virtual Machine Monitor (VMM), que es el encargado de la gestión de recursos y peticiones de cada uno de los dominios o sistemas operativos hacia el hardware real.

a) Hipervisor Xen

El Xen hipervisor es el corazón de Xen, es lo primero que ejecuta el gestor de arranque GRUB[11] y se encarga de controlar el hardware (CPU, memoria, etc.) y distribuir su uso entre las diversas máquinas virtuales. Se inicia por debajo del sistema operativo anfitrión (dom0), proporcionando estabilidad, aislamiento entre máquinas y políticas de calidad de servicio (QoS), por estos motivos necesita correr en un entorno privilegiado.

Los procesos a ser gestionados por el hipervisor son los siguientes:

"Gestión de memoria:

- *Segmentación: no se pueden usar descriptores de segmentos con todos los privilegios y tampoco se superponen los segmentos con el final del espacio de direcciones.*
- *Paginación: el sistema operativo invitado tiene acceso directo a las tablas de paginación, pero para actualizarla debe ser validado por el hipervisor.*

CPU:

- *Protección: el sistema operativo invitado debe correr en un nivel de privilegios menor que el hipervisor.*
- *Excepciones: el SO invitado debe registrar una tabla de manejadores de excepciones en Xen, de tal manera que, por ejemplo, las faltas de página las ejecute el hipervisor.*

[11]GRUB: (GRand Unifier Bootloader) es un gestor de arranque linux. Es lo primero que se carga cuando se inicia un computador (booteo) y permite tener diferentes sistemas operativos, y diferentes versiones de ellos, instalados en un mismo disco duro.

- *Llamadas al sistema: las llamadas al sistema se ejecutan directamente, para esto previamente se deben validar, de tal manera que se mantenga el aislamiento entre máquinas virtuales.*
- *Interrupciones: las interrupciones se reemplazan por eventos del sistema.*
- *Tiempo: cada máquina virtual tiene una interfaz de tiempo, para mantener la diferencia entre el tiempo real y el tiempo virtual."* (Olcina, 2008)

Dispositivos de E/S:

En el caso de los dispositivos virtuales, para poder trabajar con estos se capturan sus interrupciones hardware y se sustituyen por un mecanismo de eventos.

El hipervisor Xen provee una interfaz de dispositivos genéricos con los que interactúa. Cuando una máquina virtual utiliza un dispositivo, la orden va al controlador de esta máquina virtual (que no es más que una interfaz del controlador real que está en el sistema operativo anfitrión), aquí se traduce la petición a los drivers nativos de los dispositivos físicos y se ejecuta la orden.

Cuando se usa ***full virtualization*** (virtualización completa), se emulan completamente los comportamientos de los dispositivos de la máquina virtual, en cambio que la *paravirtualización* únicamente crea una capa de abstracción sobre los dispositivos reales.

b) *Los dominios en Xen.*

Xen maneja sus máquinas virtuales con el nombre de dominios (domX) y pueden ser de dos tipos:

- ***"dom0 (Host):*** *este dominio se lanza cuando arranca el kernel modificado de Xen, el anfitrión, aquí arranca el demonio xend que es el que inicia todos los procesos, la máquina con privilegios desde la que se crean/arrancan/eliminan/migran las otras máquinas virtuales o dominios. Aquí se gestionan las tareas de administración al hipervisor (Xen Monitor) y tiene acceso directo al hardware físico del host y se proporciona la clase de dispositivos genéricos a los domU.*
- ***domU (dom1, dom2,…domN):*** *son las demás máquinas virtuales, los denominados hosts invitados. Se forman con el driver de dominio que será el encargado de administrar los dispositivos asignados (backend) a la nueva máquina virtual, quitando responsabilidades al dom0 haciendo el sistema más estable y haciendo creer a cada sistema operativo que se ejecuta sobre ese hardware genérico."* (Wiki XenSource, 2010)

c) *Interfaces de red XEN*

Xen crea pares de interfaces Ethernet virtuales interconectadas para que dom0 las utilice. Se pueden concebir como dos interfaces Ethernet conectados por un cable Ethernet cruzado interno. La

interfaz veth0 está conectada a vif0.0, veth1 está conectada a vif0.1, y así sucesivamente.

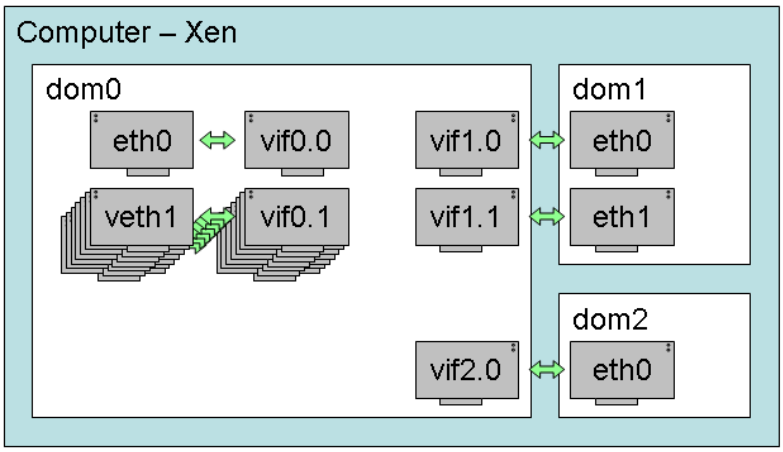


Figura 12. Interfaces de red virtuales Xen. **(tldp.org, 2009)**

Cada vez que se crea una instancia domU, ésta recibe un identificador numérico (asignado automáticamente y sin la posibilidad de que el usuario lo elija). El primer domU será el número 1, el segundo el número 2, incluso aunque el número 1 ya no se esté ejecutando, etc. Para cada nuevo domU, Xen crea un nuevo par de interfaces Ethernet virtuales conectados, con un extremo de cada par dentro del domU y el otro en el dom0. Si el domU usa Linux, el nombre de dispositivo se mostrará como eth0. El otro extremo de ese par de interfaces ethernet virtuales aparecerá dentro del dom0 como interfaz vif#.0. Por ejemplo, la interfaz eth0 del domU número 8 está conectada a vif8.0. Si se crean múltiples interfaces de red dentro de un domU, sus extremos se verán como eth0, eth1, etc., mientras que dentro de dom0 aparecerán como vif#.0, vif#.1, etc.

Cuando un domU se detiene (comando “xm shutdown <domId”>), las interfaces ethernet virtuales que se crearon son eliminadas.

- **El script network-bridge.**

En el momento en que Xen arranca, ejecuta el script /etc/xen/scripts/network-bridge, el cual lleva a cabo las siguientes tareas:

  - *"Crea un nuevo puente llamado xenbr0 (pudiéndose cambiar este nombre).*
  - *Desactiva la interfaz Ethernet real eth0.*
  - *Copia las direcciones MAC e IP de la eth0 a la interfaz virtual de red veth0.*
  - *Renombra la interfaz real eth0 a peth0.*
  - *Renombra la interfaz virtual veth0 a eth0.*
  - *Conecta peth0 y vif0.0 al puente xenbr0.*
  - *Activa el puente, peth0, eth0 y vif0.0."* (Olcina, 2008)

- **El script vif-bridge**

En el caso de las máquinas virtuales, en el momento de arrancar un domU, xend, que se está ejecutando en dom0, lanza el script vif-bridge, el cual lleva a cabo las siguientes tareas:

  - Enlaza la interfaz vif#.0 al puente xenbr0.
  - Levanta la interfaz vif#.0.

Cuando vamos a implementar la plataforma Xen en un equipo hay que tomar en cuenta que es preferible contar con dos interfaces de red. Esto servirá para que la primera sirva para controlar el dominio anfitrión Dom0 e implementar las políticas de acceso y la segunda interfaz para la interacción de las máquinas virtuales o

domUs con el exterior. De esta manera se asegura que no existan interferencias por la presencia del firewall en el dom0 en el caso de contar con una sola interfaz de red física.

d) Redes Virtuales en XEN

En XEN se pueden configurar redes virtuales dentro de los propios dominios. Entre las más habituales configuraciones de red tenemos:

a) Puente (bridge).
b) Encaminador (router).
c) NAT.

El archivo para habilitar cada una de estas configuraciones está en:

/etc/xen/xend-config.sxp [Ver Anexo F]

- **Herramienta brctl.**

BRCTL hace referencia a una herramienta de red que inspecciona y modifica la configuración de los puentes o bridges. Esto nos será muy útil en el momento de verificar las conexiones de red asociadas a cada uno de los dominios o máquinas virtuales además de proporcionarnos información como las direcciones MAC que ya han sido utilizadas en cada una de las interfaces Ethernet tanto físicas como las virtuales.

Se deben tener privilegios de superusuario o root, antes de ejecutar estos comandos.

Tabla 16. Comandos generales de brctl.

brctl show	Muestra la información de los puentes y las tarjetas que están conectadas al puente.
brctl showmacs xenbr0	Muestra la información de las MAC que tiene el puente xenbr0, también indica que direcciones MAC ya están ocupadas.
brctl addbr <nombre puente>	Crea un puente o bridge.
brctl addif <puente><interfaz>	Asocia un puente a una determinada interfaz de red.

3.1.2 Migración de máquinas virtuales.

Los domUs o máquinas virtuales Xen pueden ser migradas normalmente primero apagando el dominio en ejecución para su traslado y también en caliente (live migration) entre equipos físicos sin pararlos, donde primeramente la máquina virtual origen y la máquina virtual destino negocian los requisitos de hardware para cerciorarse de que serán suficientes para la instancia en cuestión. Durante este proceso, la memoria de la máquina virtual es copiada iterativamente al destino sin detener su ejecución. Xen toma un tiempo de pausa muy breve, alrededor de 60 a 300 milisegundos, para realizar la sincronización final antes de que la máquina virtual comience a ejecutarse en su destino final. Una tecnología similar es utilizada para la migración en frío en donde se suspende la máquina virtual (comando -xm save-) en el cual se guarda un estado o imagen del equipo virtual hasta ese momento, para que posteriormente pueda ser restaurada (comando xm restore) en el disco de otra máquina host o dom0. Las conexiones de red que puedan existir en la máquina virtual migrada no son interrumpidas, una vez que ha migrado el equipo virtual, sus conexiones son re-enrutadas en el nuevo entorno en el que se ejecuta, incluyendo la redirección de las conexiones establecidas actuales o futuras de la antigua localización a la nueva, dando una transparencia óptima e inapreciable a los servicios en ejecución del dominio guest (hay que tener en cuenta que deben ser del mismo segmento LAN).

3.3 Herramientas para la administración de infraestructuras virtuales.

3.3.1 OpenQRM

Figura 13 Logo OpenQRM. **(OpenQRM, 2010)**

"OpenQRM es una plataforma open source para recolectar datos que cubre muchas facetas de administración. Provee una única consola de administración para la infraestructura IT[12] completa de la organización y una buena API que puede ser usada para integrar herramientas de terceras partes.

Esta desarrollado con Java, C, Javascript, Perl, PHP y shell's Unix. Usa como backend a MySQL y PostgreSQL." (Arias, 2009)

Actualmente openQRM está en la versión 4.6 y su interfaz de administración es muy intuitiva. Una desventaja de esta herramienta es que la documentación disponible en su página web no es muy específica y no revela muchos detalles en cuanto a parámetros de configuración con equipos host ya pre-instalados y elementos de post-configuración.

Funciona con una variedad de plugins (los más básicos son DHCP, TFTP, y LocalServer) los mismos que una vez iniciados funcionan muy bien para la gestión de recursos disponibles localmente. Para administrar recursos virtuales es necesario activar los plugins mencionados anteriormente más los plugins de Xen y Storage-Xen.

[12]IT: Tecnología de la información.

3.3.2 Convirture

Figura 14. Logo de Convirture. **(Convirture, 2011)**

Convirture se ha convertido en una potente herramienta de administración de entornos virtualizados y una de las más desarrolladas en los últimos años, especialmente enfocada a las plataformas Open Source. Ha tenido una evolución significativa en funcionalidades desde su primera versión, tanto así que ahora también soporta la gestión de virtualización en KVM, ampliando su campo de gestión en tecnologías de virtualización Open Source.

Entre las características de este administrador tenemos las siguientes:

- Es Open Source.
- Permite la administración y configuración accesible a través de una interfaz web.
- Se monitoriza en tiempo real tanto los recursos físicos como los virtuales.
- Permite tener administradores múltiples y cada acción se registra en una bitácora de eventos.
- Incluye un repositorio de datos centralizado siendo personalizable según las necesidades de cada tecnología de almacenamiento compartido.
- Permite el agregar varios servidores en un conjunto de recursos (pool) para ser administrados de mejor forma y aprovechar ventajas como la migración de máquinas virtuales y la configuración de entornos de alta disponibilidad (HA).

3.3.3 Spacewalk

Figura 15. Logo Spacewalk. (Red hat, 2010)

Spacewalk es un administrador de sistemas de código abierto bajo licencia GPL v2 y gratis que está desarrollado por una comunidad de desarrolladores y

apoyado por Red Hat que lo utiliza en su producto Red Hat Network Satellite. Spacewalk trabaja con la versión Red Hat Enterprise Linux, Fedora, CentOS y otros derivados de esta distribución.

Entre las funciones más destacables de Spacewalk:

- “Inventario hardware y software.
- Instalar y actualizar las aplicaciones de manera centralizada.
- Recopilar y distribuir paquetes de software personalizados en grupos administrables.
- Aprovisionamiento de sistemas.
- Administrar y desplegar archivos de configuración.
- Monitorizar.
- Aprovisionar, iniciar, parar y configurar máquinas virtuales.
- Distribuir contenido a través de múltiples sitios geográficos.
- Soporte para virtualización a través de Kernel based machine.
- Soporte de arquitecturas de 32bits y 64bit”. (Red hat, 2010)

Este administrador no fue posible ser examinado en funcionalidades, ya que en la instalación se tuvo problemas de dependencias con algunos paquetes que no fueron detectados (específicamente oracle-instantclient11.2-basic y oracle-instantclient11.2-sqlplus), necesarios para enlazar la administración con Oracle (versión 11 XE) bajo el sistema operativo CentOs 5. Otro punto era la orientación de software cerrado de la base de datos Oracle, por su carácter de privativa no se enmarca en la línea del proyecto por lo que se desistió de esta herramienta.

3.3.4 Virt Manager

Figura 16. Logotipo de Virt-Manager (Red Hat Linux, 2010)

La aplicación "Virtual Machine Manager" o también llamada virt-manager se refiere a una interfaz de usuario de escritorio para administrar máquinas virtuales. Presenta una visión resumida para correr dominios u hosts, su ejecución es en tiempo real y además brinda estadísticas de utilización de los recursos. Se visualizan detalles del rendimiento en modo de gráficas y los tiempos de utilización. Además permite administrar la creación de nuevos dominios, configuración y ajustes de asignación de recursos de un dominio y hardware virtual. También integra un visor de VNC[13] integrado cliente presenta una completa consola gráfica en el dominio resultado.

A continuación se presenta una tabla comparativa de las herramientas analizadas para la administración de infraestructuras virtuales las cuales descriptivamente gozan de una para podernos dar cuenta de la notable funcionalidad de las herramientas de administración Open Source en los entornos de Cloud Computing o computación de la nube.

Tabla 17. Comparativa de las herramientas de administración Open Source de entornos virtuales.

Comparativa	openqrm	Convirture	Spacewalk	Virt-Manager
API	X	x	x	
Administración de la configuración	X	x	x	x
Interfaz web.	X	x	x	
Flexibilidad de recursos	X	x		
Integración de almacenamiento	X	x	x	x
Monitoreo	X	x	x	x
Soporte para tipos de virtualización	xen/kvm/vmware	xen/kvm	xen/kvm	xen, kvm
Seguridad	X	x	x	
Soporte	X	x	x	x
Soporte para sistemas físicos	X		x	

[13]VNC: Son las siglas en inglés de Virtual Network Computing (Computación Virtual en Red)

HA (Alta disponibilidad)	X	x	x	
Configuración de redes	X	x	x	x
Usabilidad		x		x

De las opciones analizadas, ***Convirture*** *es la herramienta seleccionada como la más adecuada para trabajar en el proyecto, ya que ofrece más flexibilidad, compatibilidad, estabilidad con la tecnología XEN que las otras opciones para la gestión y también brinda soporte para KVM que está en continuo mejoramiento, lo que permitirá en un futuro un menor impacto en el momento de que las necesidades de migración de plataforma así lo requieran.*

La ventaja de usar una herramienta de gestión web es la independencia de sistema operativo permitiendo una cómoda gestión de las operaciones básicas, aunque con cierto riesgo por lo centralizado de la administración en una sola interfaz, ya que quien accede tiene poder completo sobre la infraestructura virtual. Convirture nos permitirá gestionar, monitorear a los equipos de la infraestructura y además nos libera de la necesidad de instalar versiones de cliente en nuestro equipo de administración.

Además de esta herramienta tendremos como alternativa a ***Virt-Manager*** que también será usada en la gestión de los equipos virtuales tanto localmente así como remotamente (por ssh) a través del api de libvirt en el caso de que no se pueda conectar con la interfaz web de Convirture. Por otro lado siempre es recomendable tener una conexión de *contingencia* para lo cual virt-manager nos proveerá de la misma; por la *estabilidad* y su *ejecución nativa* adicionalmente ofrece otras funcionalidades como mayor simplicidad al momento instalar, configurar y de realizar las operaciones sobre las máquinas virtuales residentes en el anfitrión o dom0, sobre todo en el modo consola que es mucho más completo, avanzado y mucho más familiar para un administrador de servidores.

El resto de herramientas analizadas si bien soportan la gestión de máquinas virtuales, pero algunas están limitadas a realizar operaciones básicas de control y monitoreo. Al mismo tiempo están encaminadas a gestionar un gran número de hosts y máquinas virtuales, punto que limitó el explorar sus funcionalidades completamente por las características de nuestro entorno de pruebas con grandes sistemas de almacenamiento avanzado especialmente orientado a trabajar bajo entornos actualmente en auge como son los clusters, Grid y Cloud Computing.

CAPÍTULO IV

IMPLEMENTACIÓN Y EVALUACIÓN DE LA SOLUCIÓN CON LOS SERVICIOS MENOS CRÍTICOS.

4.1 Instalación y/o configuración de servidores virtuales

Para la presente fase de implementación de la plataforma, se han tomado en cuenta varios puntos importantes, los mismos que exponemos a continuación:

- Arquitectura y esquema de red (externo e interno).
- Requerimientos de hardware.
- Requerimientos de software.
- Monitoreo de los equipos virtuales.
- Análisis y evaluación de desempeño de los equipos virtuales.
- Evaluación de la seguridad.

4.1.1 Arquitectura y esquema de red externo e interno.

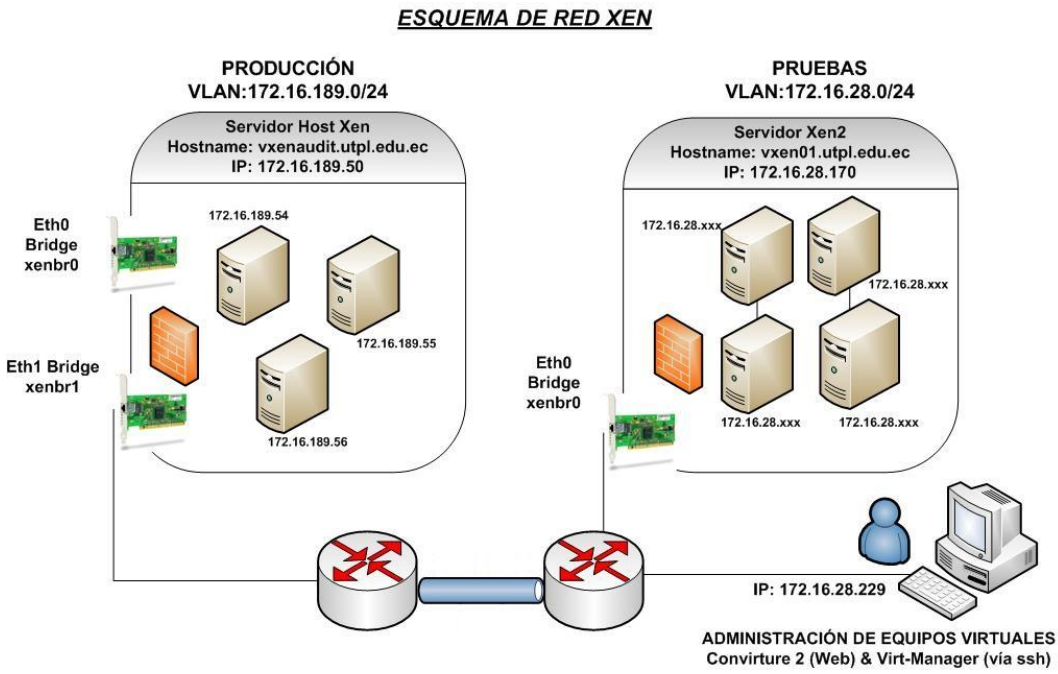


Figura 17. Esquema de servidores XEN.

En la figura anterior se muestra el esquema de red con que funcionarán nuestros servicios virtualizados en el que intervienen dos servidores (dom0) y una red NFS[14] para compartir los recursos y las imágenes para

[14] NFS: (Network File System) Posibilita que distintos sistemas conectados a una misma red accedan a ficheros remotos como si se tratara de locales.

la migración de máquinas virtuales. Se ha establecido de manera que cada una de las máquinas virtuales actúe cada una como un host independiente, por lo que están totalmente aisladas una con respecto de otra.

En la configuración de red, existe la posibilidad de tres métodos de configuración (bridge, NAT y router) cuyo funcionamiento fue descrito en el capítulo II.

Puesto que los servicios a implementar van a funcionar cada uno, con una interconexión indirecta entre equipos virtuales, aquí aplicamos la configuración bridge (puente) para permitir que los equipos virtuales salgan a través de la interfaz del dom0 u Host, hacia el exterior y se pueda establecer la conexión hacia los mismos de manera autónoma.

Una vez instalado Xen, lo que sigue es editar el archivo de configuración de la misma ubicado en **/etc/xen/xend−config.sxp**.

Nos interesa crear un puente Ethernet que estará formado por una o más interfaces Ethernet virtuales para cada máquina virtual y todas ellas unidas a la interfaz real. Como en nuestro caso contamos con solo una interfaz de red, la configuración es relativamente sencilla; pero se puede acoplar a cualquier número de tarjetas de red tan solo hay que personalizar los scripts en */etc/xen/scripts,* y acoplarlo según nuestras necesidades.

a) Preparando el entorno de red Xen.

Hay puntos a tomar en cuenta para preparar el entorno de red Xen y que nos permitirán evitar algunos problemas en su configuración y correcto funcionamiento:

- Xen3 soporta hasta 8 interfaces virtuales por domU.
- En la configuración de red, Xen crea la interfaz virbr0, que sirve como enlace para realizar NAT. Se puede desactivar esta interfaz removiendo los links simbólicos existentes en /etc/libvirt/qemu/networks/autostart.

- Los dispositivos virtuales son provistos por el controlador de netloop y solamente admite ocho dispositivos virtuales a la vez. Este valor puede ser levantado incrementando el valor del parámetro de nloopbacks:

  # modprobe netloop nloopbacks=128
- Existe algunos problemas con los números mib en el monitoreo snmp, ya que estos cambian en cada reinicio el domU.
- Otra buena práctica es nombrar distintivamente a las interfaces vifx.y en el archivo de configuración para que no cambien estos valores en cada reinicio.

  vif=['vifname=dnscache', 'vifname=bt4']

  Para que se aplique la configuración de interfaces anterior se necesita apagar el domU. A veces no es suficiente con reiniciar.
- Los nombres a asignar vif solo acepta hasta 15 caracteres por lo que hay que tener en cuenta esta configuración en el caso de identificadores muy largos.
- Es preferible asignar una dirección de IP y MAC al equipo virtual antes de iniciarlo, para que XEN aplique las reglas de antispoofing evitando este tipo de ataque y problemas en la conexión por la aplicación de estas reglas.

- Al cambiar una dirección MAC debemos limpiar la cache out udev's. Para lograr que no se detecte como una tarjeta adicional al eth0. El archivo esta en /etc/udev/rules.d/z25_persistent-net.rules. De esta manera al reiniciar nos reconoce como una interfaz eth0 nueva y no nos genera una nueva MAC aleatoria. Se puede activar DHCP en el archivo de configuración agregando las líneas:

  dhcp=1

- Para el mecanismo de seguridad que maneja XEN, llamado antispoofing, funciona tanto para la configuración de network-bridge como para la de network-route, mas no trabaja con network-NAT.

Para esto agregamos en el archivo de configuración xend-config.spx, la sentencia:

(network-script 'network-bridge antispoof=yes')

- La dirección IP se puede especificar connotación CIDR(Classless Inter-Domain Routing)

  vif=['ip=10.0.0.1/8']

- Cuando hay problemas de conectividad se puede agregar al archivo /etc/xen/scripts/networkbridge la linea:

  echo 1 > /proc/sys/net/bridge/bridge-nfcall-iptables

- Otro problema sucede con las FORWARD chains, cuando tienen exceso de carga puede haber pérdida de paquetes. Una solución es agregaren el archivo de vif-bridge, buscando la funcion frob_iptable(), agregando la línea:

iptables -t raw "$c" PREROUTING -m physdev --physdev-in "$vif" "$@" -j NOTRACK

- De preferencia trabajar sin la aplicación Network Manager porque en algunas distros causa mal funcionamiento de network-bridge.
- Activar STP (Spanning Tree Protocol) ayuda previniendo bucles en la red y puede ser activado en el caso de tener redes complejas con muchos bridges.
- Se puede renombrar el puente de red por defecto por otro y además especificar con que interfaz de red se trabaja:
  (network-script 'network-bridge bridge=xenbr0 netdev=eth1')
- Para verificar el tráfico con la interfaz física:
  #tcpdump -n -i peth0
- Verificar las configuraciones de red estáticas y los dns de cada domU:
  /etc/sysconfig/network-scripts/ifcfg-eth0
  /etc/resolv.conf

También existen otras opciones, en el caso de querer que nuestras máquinas virtuales estén en redes diferentes, por ejemplo 192.168.189.x para dom0 y las demás máquinas físicas de la red y 10.0.0.x para las máquinas virtuales, y luego usar DNAT con iptables para enrutar el tráfico, necesitaremos usar las directivas (network−script network−nat) y (vif−script vif−nat).

Para cada máquina virtual creada, XEN crea un nuevo par de interfaces Ethernet virtuales conectadas, con un extremo de cada par dentro del domU y el otro en el dom0. Si el domU usa Linux, el nombre de dispositivo se mostrará como eth0. El otro extremo de ese par de interfaces Ethernet virtuales aparecerá dentro del dom0 como interfaz vif<#>.0. Por ejemplo, la interfaz eth0 del domU número 5

está conectada a vif5.0 y así de tal forma que si se crean varias interfaces de red dentro en un domU, sus extremos se verán como eth0, eth1, eth2, etc, mientras que dentro de dom0 aparecerán como vif#.0, vif#.1, etc. En la siguiente figura se ilustra esta configuración:

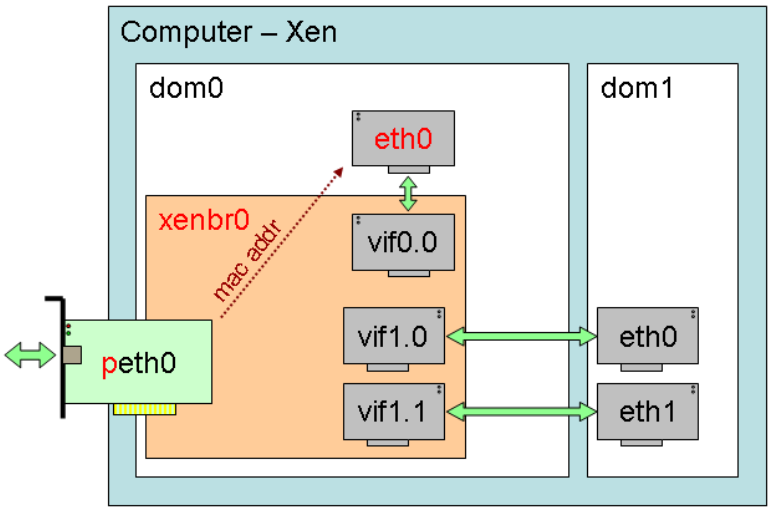


Figura 18. Esquema de red en modo bridge. **(tldp.org, 2009)**

A. Especificación del hardware

El hardware a manejar en la fase de implementación del presente proyecto tiene las siguientes características:

EQUIPO SERVIDOR XEN 1

- Equipo Lenovo modelo MT-M8910
- Procesador Intel Quad Core2 de 2.0 Ghz con tecnología VT habilitada.
- Memoria RAM de 4 GB
- Disco duro de 320 GB
- 1 tarjeta Ethernet 10/100 Mb.

EQUIPO SERVIDOR XEN 2

- Equipo IBM Modelo XSeries.

- Procesador Intel Xeon de 1.6 GHZ con tecnología VT habilitada.
- Memoria RAM de 3GB.
- Disco de 30GB.
- 2 tarjetas de red Gigabit Ethernet 100/1000 Mb.

Debido a que uno de los puntos débiles de las virtualización es la existencia de un punto central de fallo, hay recomendaciones que se deben tomar en consideración para mitigar dicho riesgo. Para ese problema hay que tener como requerimiento que el hardware cuente con redundancia en los dispositivos más críticos como son: procesador, memoria RAM, discos duros en RAID al menos en nivel 1 y que estos tengan la propiedad de ser reemplazados en caliente (llamados también dispositivos hotspare). Para entornos de producción más crítica se recomienda implementar servicios de alta disponibilidad (HA) y tecnología basada en clusters.

B. Especificación del software.

Para el servidor anfitrión:

- Sistema Operativo Centos 5.5 de 64 bits, con todos los paquetes del grupo “Virtualization” y SSH habilitado para la administración remota.

Para el servidor de administración:

- Sistema Operativo Ubuntu 10.04 (32 o 64 bits) con los paquetes de “virt-manager” instalados y servicio ssh habilitado.
- Paquetes de Convirture2 descargados. (convirt-install-2.0.1.tar.gz, convirt-2.0.1.tar.gz, convirture-tools-2.0.1.tar.gz).

Luego en la parte de almacenamiento, usamos las siguientes particiones en la máquina anfitrión o dom0 con Centos 5.4 (dom0):

• /boot 150 MB (ext3)

• /vm el resto del disco en un LVM (grupo de volúmenes lógicos 250GB)

Un volumen lógico llamado swap en /swap de 3000 MB

Un volumen lógico llamado root en / (directorio raíz) de 50 GB

Para la instalación de los paquetes del hipervisor de Xen V. 3.0.18, son necesarios los siguientes requerimientos:

- *Make*
- *GCC*
- *Librerías de desarrollo de curses*
- *Python, no solo el intérprete sino el paquete de desarrollo necesario para compilar algunos módulos de Xen.*
- *Latex, para generar la documentación*
- *Bridge utils*
- *Iproute*

Para la instalación de la plataforma base XEN, se la realizará usando el sistema operativo Linux CentOs 5.5, debido a que esta distribución ofrece funcionalidades de administración muy reconocidas entre los administradores de servidores, y es considerada como una de las distros libres más estables para entornos de producción y sobre todo para implementar virtualización Open Source.

Otro aspecto importante es el manejo del almacenamiento en las máquinas virtuales. Para este caso manejaremos particiones LVM en cada servidor físico, porque ayuda a manejar dinámicamente el aumento o disminución de demanda de este recurso.

Para poder crear manualmente los discos duros virtuales en donde van a residir los sistemas operativos usamos el comando ***dd***. La notación es la siguiente:

dd if=/dev/zero of=/virtual_servers/owasp.img oflag=direct bs=1M seek=4099 count=1

Tabla 18. Parámetros del comando dd.

PARÁMETRO	DESCRIPCIÓN
if=/dev/zero	Lee desde la unidad infinita de ceros.
of=/virtual_servers/owasp	Escribe en la ruta /virtual_servers/owasp
oflag=direct	Escribe datos con comas.
bs=1M	Lee y escribe bytes a 1 MB.
seek=4009	Tamaño final del bloque.
count=1	Copia bloques de tamaño en bytes.

C. Monitoreo de los equipos virtuales.

El monitoreo de los equipos virtuales se los realizará a través de la herramienta incluida en *Convirture* que nos proporciona en modo gráfico la carga en los host físicos y virtuales en tiempo real y en donde cada proceso es registrado en una bitácora de tareas.

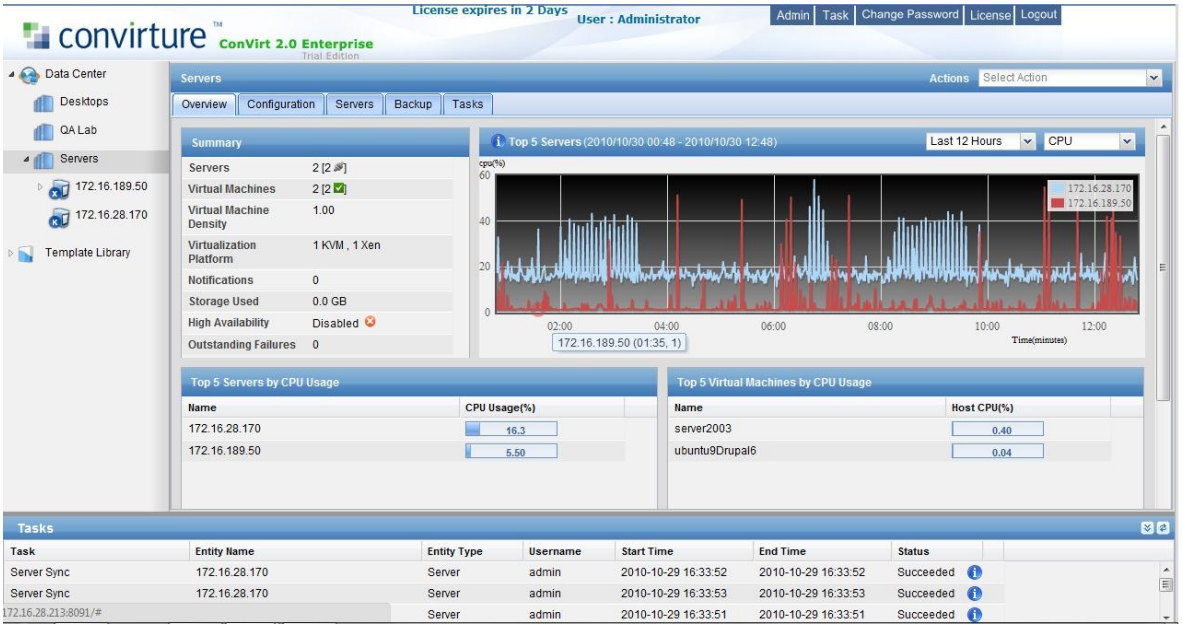


Figura 19. Interfaz de administración web en Convirture

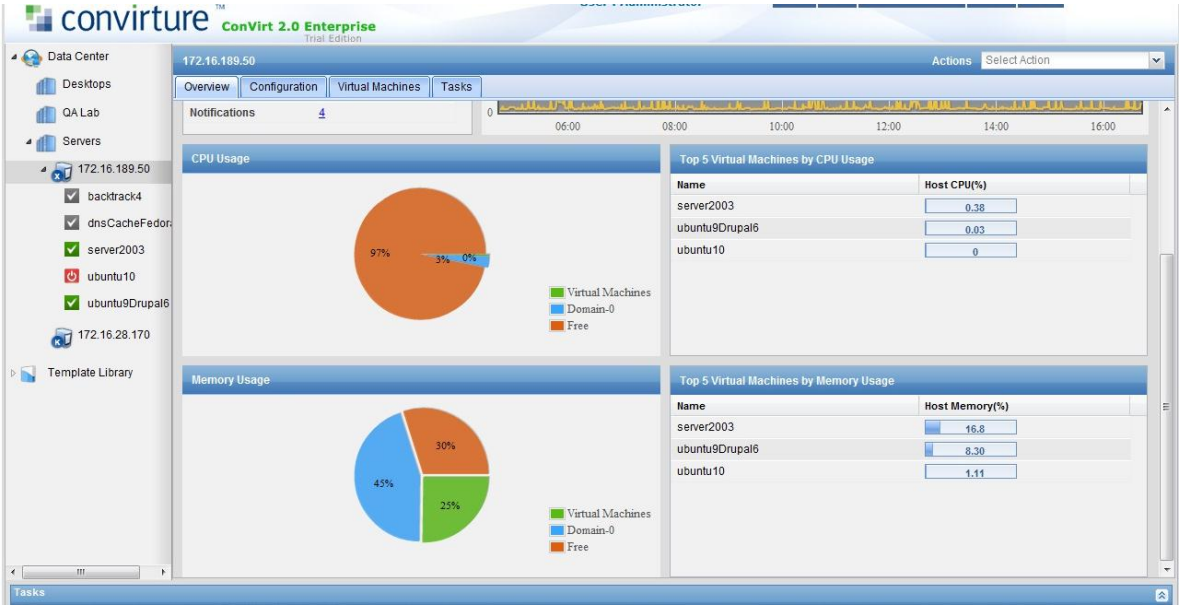


Figura 20. Gráfica de rendimiento en la interfaz web de Convirture.

Otra herramienta utilizada en el monitoreo es *Munin*, la misma que nos permitirá examinar el rendimiento de manera más detallada de cada elemento individual de los equipos virtuales en donde podemos ver datos importantes como es el análisis de carga en procesador, memoria, red de datos y algunos otros dispositivos temporizados en términos de día y semana.

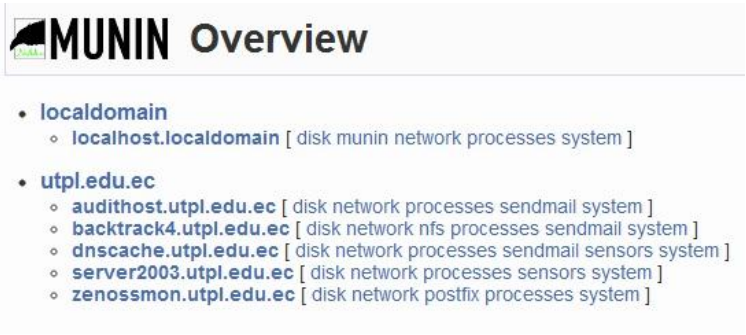


Figura 21. Interfaz web de la herramienta de monitoreo Munin.

La utilidad *xm,* incluida por defecto en cada host o dom0 también es muy útil para realizar este proceso de monitoreo de carga y rendimiento.

Por consola desde una conexión ssh[15] desde el dom0 podemos revisar cuantas máquinas virtuales están corriendo mediante el comando:

#xm list

Name	ID	Mem (MiB)	VCPUs	State	Time(s)
Domain-0	0	2048	1	r-----	1906.6
vowasp	3	512	1	r-----	137.9
bactrack	2	512	2	------	127.6
server2003	5	1024	1	r-----	142.5

Y para obtener una estadística más detallada de los dominios que están ejecutándose tecleamos:

#xm top

Ahora podemos acceder a la consola de cualquiera de las máquinas con solo el nombre o el id del domU.

 # xm console 3 o # xm console vowasp

En el caso de que se desee configurar que una determinada máquina virtual inicie automáticamente al arrancar el sistema base Xen, se digita en la línea de comandos:

 # ln -s /etc/xen/vm01 /etc/xen/auto

A continuación se encuentran los comandos más importantes de Xen:

Tabla 19. Comandos xm de xen.

COMANDO	DESCRIPCIÓN
xm create -c <archivo-configuracion>	Inicia la máquina virtual
xm shutdown <name>	Detiene la máquina virtual.
xm destroy <name>	Detiene la máquina virtual de golpe. Similar a la acción de presionar el botón de encendido.

[15] SSH: siglas en ingles de Secure Shell Hash que es un protocolo para asegurar la comunicación en una red.

xm list	Muestra todos los sistemas en ejecución.
xm console <name>	Inicia sesión en la máquina virtual.
xm help	Muestra la ayuda de este comando.

Se puede iniciar la creación de un equipo virtual a través de la herramienta de administración web de Convirture.

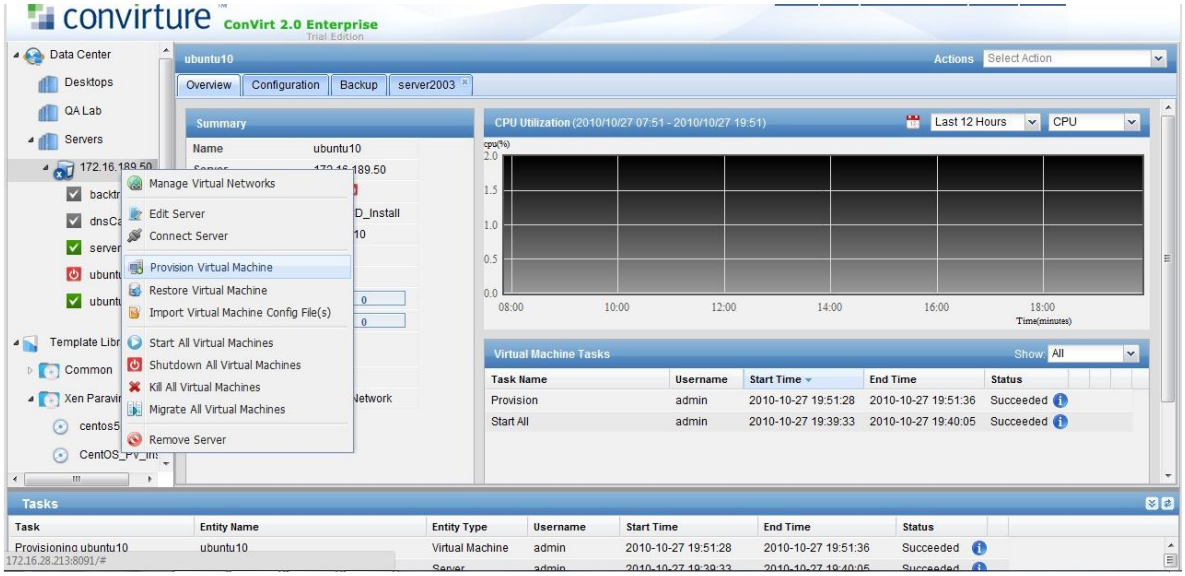


Figura 22. Creación de una VM en Convirture.

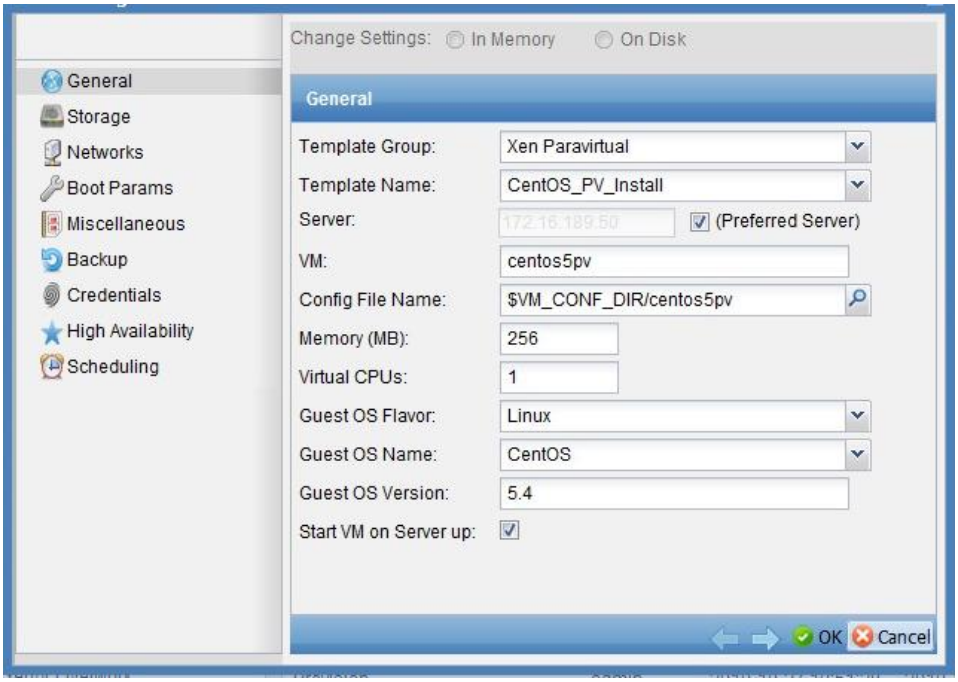


Figura 23. Especificación de parámetros de máquina virtual.

Figura 24. Especificación de dispositivos de almacenamiento virtuales.

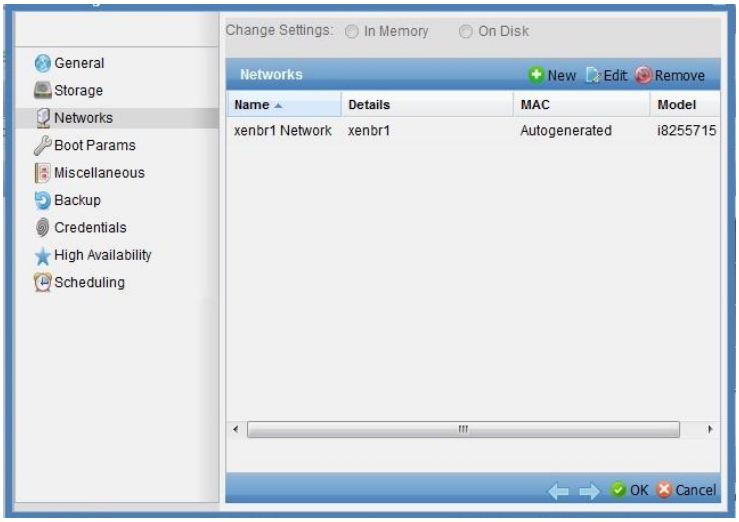


Figura 25. Especificación de parámetros de red.

También se lo puede realizar desde el cliente de escritorio de la herramienta virt-manager conectándonos remotamente con el protocolo SSH o por XML-PRC.

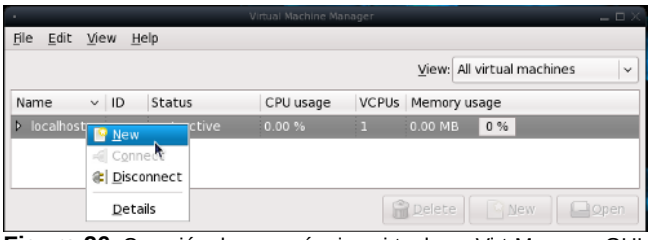


Figura 26. Creación de una máquina virtual con Virt-Manager GUI.

Aquí podemos seleccionar el archivo .ISO de un directorio o simplemente instalarlo desde una unidad óptica CD/DVD-ROM.

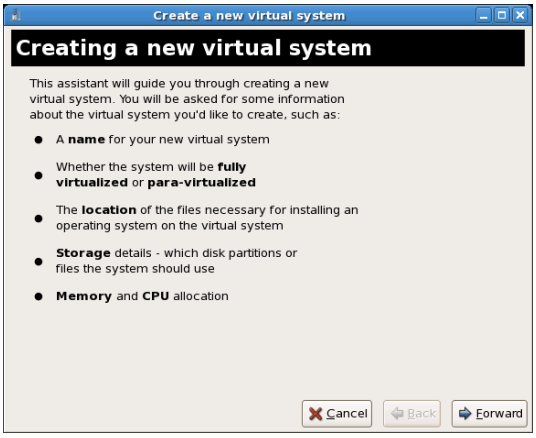


Figura 27. Parámetros necesarios para crear un nuevo DomU en Virt-Manager.

Aquí nos da los parámetros que necesitaremos especificar para la creación de un nuevo dominio virtual.

Luego tendremos que especificar el origen de instalación del nuevo equipo. En este caso puede tratarse de una imagen .iso o la unidad de CD-ROM o DVD ROM.

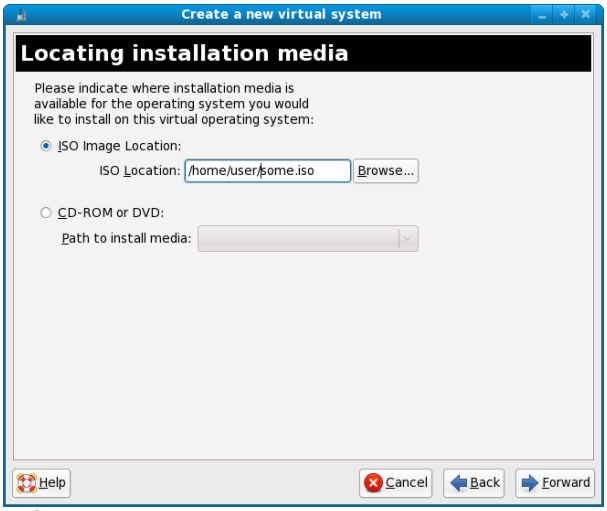


Figura 28. Especificación de medio de instalación para virtualización completa.

Finalmente tenemos los dominios instalados y operando en el servidor XEN.

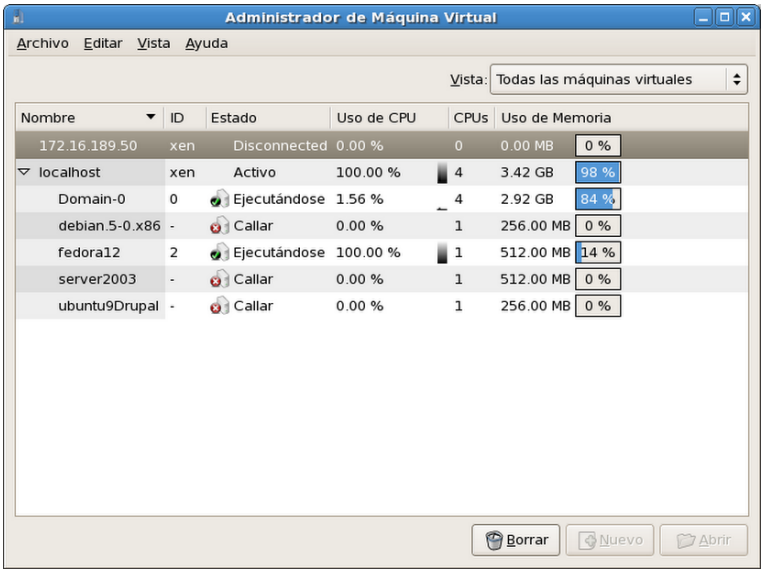


Figura 29. Servidores virtuales activos en el Dom0.

También se puede acceder vía SSH, mediante consola con la utilidad virt-install digitando el comando:

virt install - - prompt

What is the name of your virtual machine? <-- vowasp01

How much RAM should be allocated (in megabytes)? <-- 512

What would you like to use as the disk (path)? <-- /virtual_machines/owasp01.img

How large would you like the disk (/vm/owasp01.img) to be (in gigabytes)? <-- 8

Would you like to enable graphics support? (yes or no) <-- yes

What is the install location? /iso_images/owaspSLAX.iso

D. Evaluación de desempeño de los equipos virtuales.

Aquí se incluirá un monitoreo de rendimiento (diario y semanal) en cuanto a tres factores:

- Flujo de input/output[16] de tráfico de red hacia el exterior.

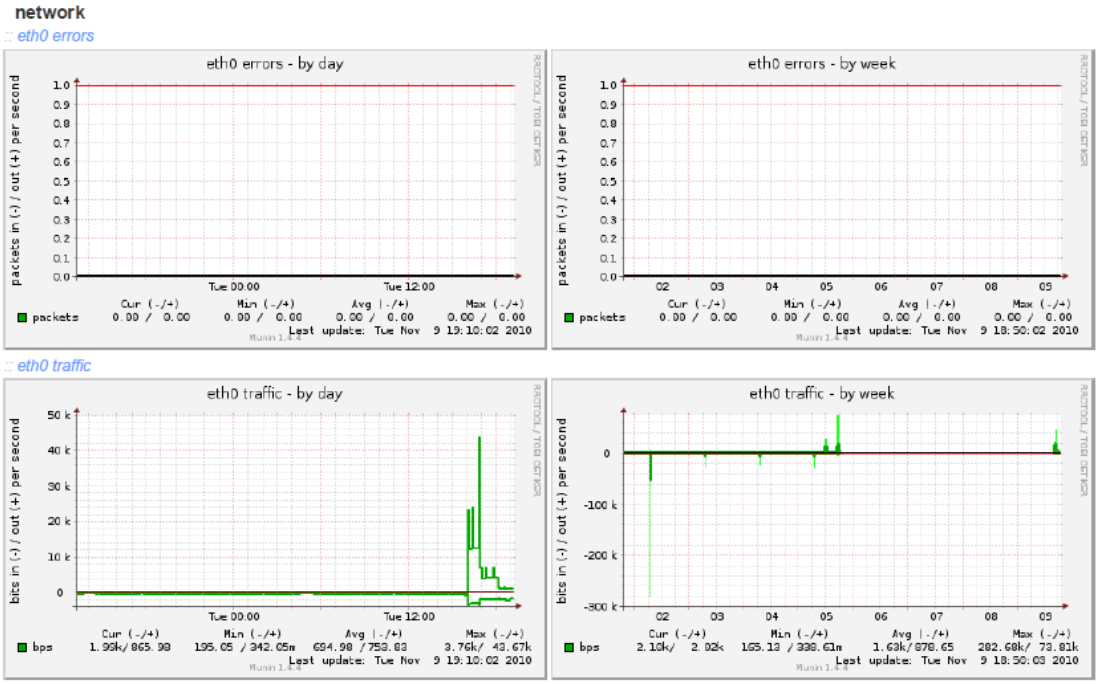


Figura 30. Monitoreo de interfaces de red con Munin.

[16]Input/Output-(Entrada/Salida) se refiere a que en ocasiones, los dispositivos o controladores de entrada y salida de datos se describen con su nombre inglés o con las siglas «I/O» en lugar de «E/S».

En este gráfico se visualiza que no hubo ningún fallo en los paquetes que se transmitían desde las interfaces virtuales y que el tráfico mayoritariamente mínimo fluía con normalidad.

- Flujo de input/output de los dispositivos de almacenamiento.

Figura 31. Monitoreo de dispositivos de almacenamiento con Munin.

La interacción de los discos duros virtuales también se visualiza como regular en donde existen algunos incrementos propios de la lectura y escritura en los discos de cada servicio pero que no superan los índices regulares de funcionamiento.

- Porcentaje de uso de procesador total.

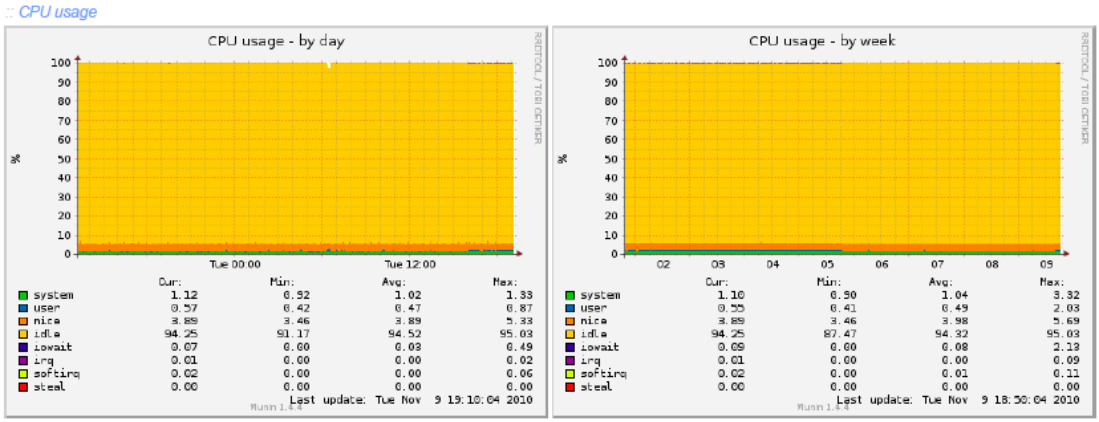


Figura 32. Monitoreo del procesador con Munin.

El procesamiento se dio sin problemas aunque se ocupaba casi la totalidad de la capacidad del procesador por la cantidad de equipos virtuales funcionando sobre el mismo procesador, el mismo se vio incrementado ya que se usa virtualización completa en algunos de los servicios por lo que esta emulación completa penaliza mucho más el rendimiento que un equipo paravirtualizado.

- Porcentaje y rendimiento de memoria RAM[17].

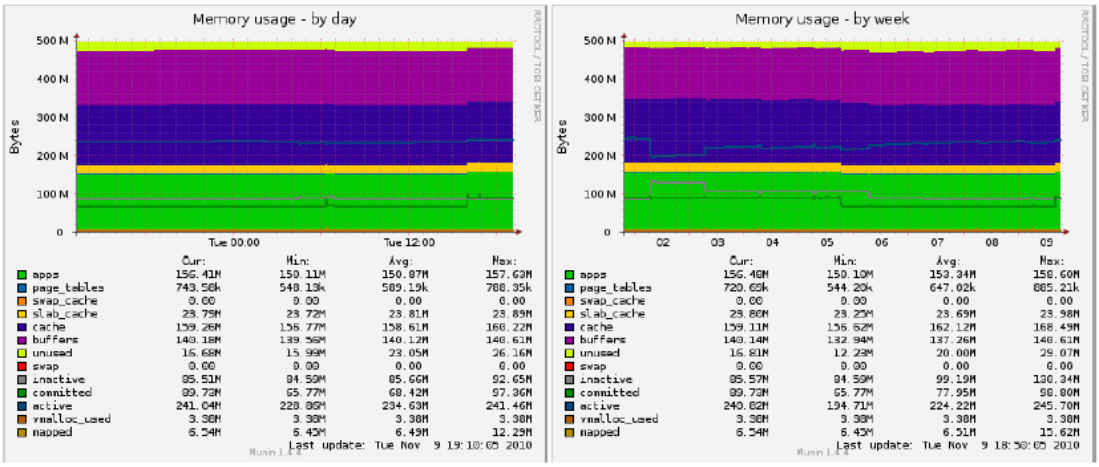


Figura 33. Monitoreo de memoria RAM con Munin

[17] RAM (Random-access memory) Son las siglas en inglés de Memoria de acceso aleatorio.

La memoria como observamos también funcionaba en porcentajes altos entre el 90 y 100% ya que los equipos se configuraron con memoria un tanto limitada (alrededor de 512MB por host) por las capacidades del host anfitrión aunque con el servicio totalmente funcional.

- Consumo de recursos del host anfitrión.

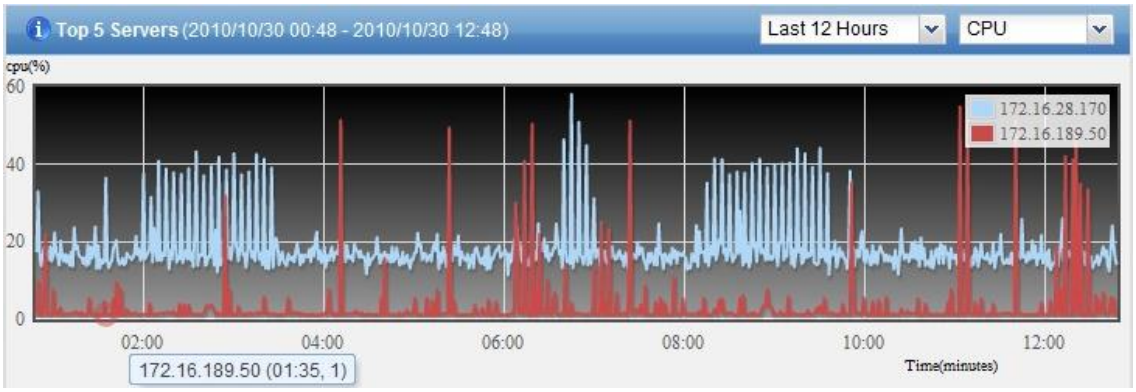


Figura 34. Monitoreo comparativo entre servidor de pruebas y el de producción.

Se logro comprobar que en el host anfitrión hubo una utilización entre el 30 y 60 por ciento de la capacidad del procesador cuando los servicios virtuales corrían en modo normal de funcionamiento sin ninguna sobrecarga en alguno de los dispositivos. En el momento de transferir archivos de gran tamaño el rendimiento incrementó hasta aproximadamente el 80 y 95% con una cierta degradación del rendimiento en cada servicio ya que hubo ligera ralentización en la respuesta a peticiones de los otros hosts virtuales.

Con esto concluimos que la implementación fue exitosa y que los servicios instalados están ejecutándose de forma estable y funcional y que su rendimiento tan solo está limitado por las capacidades actuales del hardware que cumple el mínimo recomendado.

E. Análisis de la seguridad

La seguridad es un punto importante en este tipo de implementaciones, ya que se está exponiendo el acceso y gestión total a diferentes servicios operativos en un mismo equipo físico. Por tal razón se han tomado algunas precauciones para mitigar los riesgos:

- En el dom0 o sistema operativo host se procura minimizar la instalación de paquetes innecesarios, solo la ejecución por consola está habilitada.
- La seguridad será implementada en cada host virtual según las necesidades del servicio.
- El acceso al host físico está habilitado para ejecutar instrucciones por el protocolo SSH en caso de ser necesario se use este tipo de conexión. Las opciones de XML-RPC también están habilitadas para que interactúen con el administrador Convirture. Por ende el firewall estará habilitado y solo permitirá la escucha en el puerto 22.
- En las opciones de migración o relocación de un servidor virtual están deshabilitadas y solo en caso de ser necesario se activarán.
- Generalmente en entornos de servidores Linux se usa tan solo conexión de línea por comandos, pero a menos que se requiera se tendrá que instalar y configurar servidor VNC[18] para el acceso remoto a la interfaz de dicho host virtual protegido por las restricciones de IP(s) y contraseña en el host anfitrión y en el caso de servidores Windows se habilitará la conexión de escritorio remoto y SDL habilitado en el archivo de configuración de la máquina virtual.

4.1.2 Implementación de los servicios específicos virtuales.

Aquí nos referiremos a los servicios a virtualizar, donde se detalla los componentes y requerimientos que cada uno de dichos servicios, que necesitan

[18]VNC: VNC es un programa de software libre basado en una estructura cliente-servidor el cual nos permite tomar el control del ordenador servidor remotamente a través de un ordenador cliente. (Wikipedia, 2009)

para ser configurados y de esta manera entren en funcionamiento. A continuación se detallan los servicios y sus requerimientos:

- **DNS Caching**:

El objetivo de un DNS caching o nameserver cache, es realizar consultas más rápidas en el servidor de nombres de dominio principal o master y guardar los resultados para una mayor eficacia.[Ver configuración en ***Anexo C***]

(Posteriormente lo agregamos al arranque por defecto en el inicio del sistema)

chkconfig named on
(o indicar en un nivel de ejecución específico)
chkconfig --level 3 named on

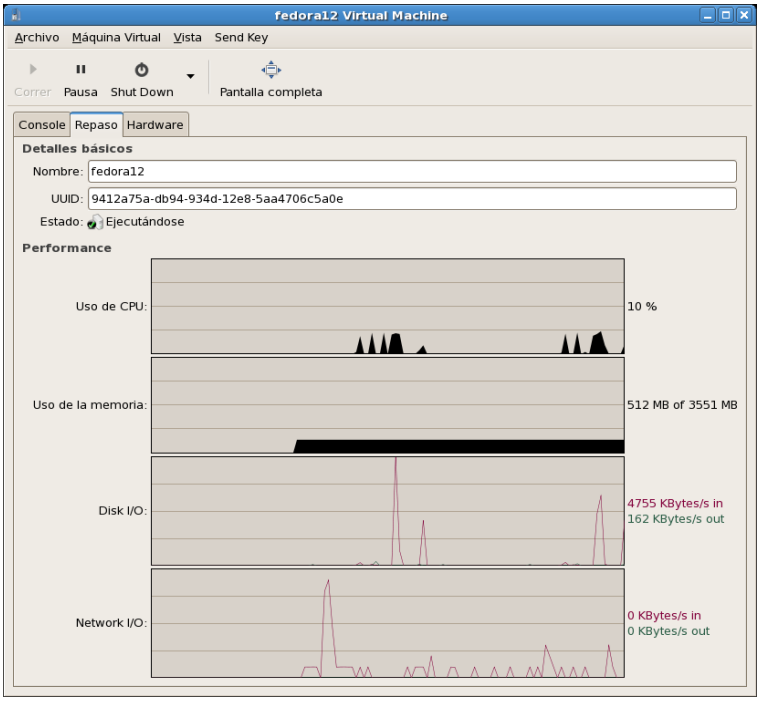
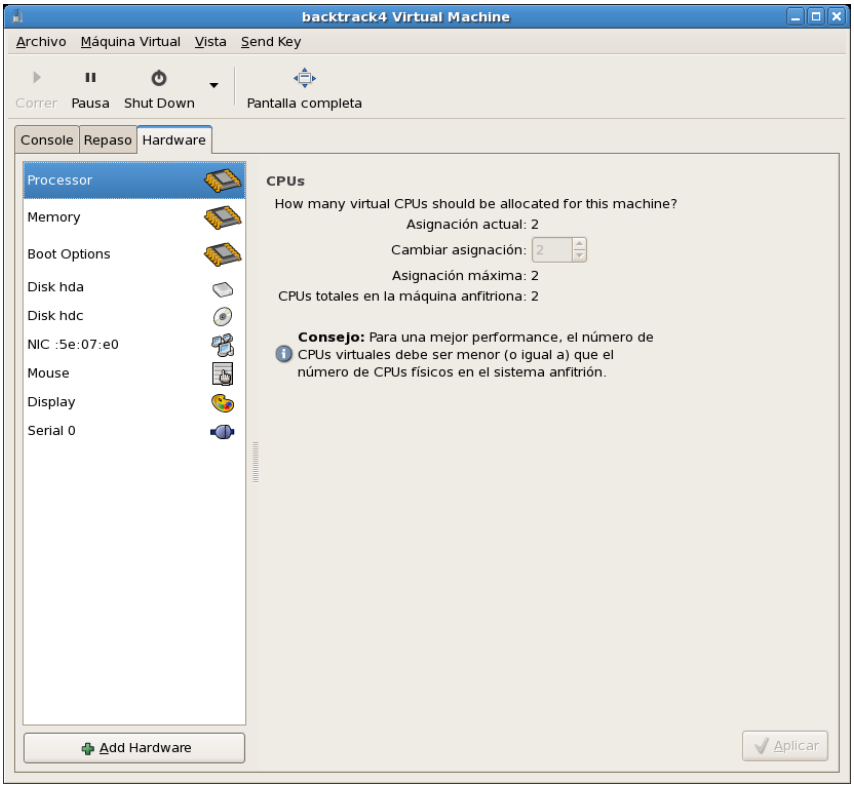


Figura 35. Gráfica de rendimiento del DomU servidor de Caching.

En la figura anterior, la herramienta virt-manager a parte de brindarnos estadísticas por consola, también nos grafica en tiempo real el rendimiento

de cada uno de los domUs y del dom0, separando en cada estadística el uso de cada componente virtual.

- **Bactrack 4**: Bactrack es una versión de linux basada en kernel y en la distribución Ubuntu, que tiene una variedad de herramientas útiles para los administradores de seguridad con herramientas para la auditoria de redes y el hacking. [Ver configuración de máquina virtual en ***Anexos***]



- **OWASP:** Es el acrónimo de Open Web Application Security Project y se refiere a un proyecto de código abierto que se especializa en determinar y combatir las causas que hacen que el software sea inseguro. Sobre todo se especializa en la auditoria de seguridad aplicada a las aplicaciones web. [Ver configuración de máquina virtual en ***Anexo C***]

Esta herramienta de auditoría está basada en un live CD que trae una compilación de algunas aplicaciones de software que nos sirven para auditar el software y nos da las pautas para que su desarrollo sea seguro.

4.1.3 Análisis de resultados.

Para las tareas de monitoreo se usaron dos herramientas. La primera incluida en la administración de *Convirture* en donde muestra con gráficas en términos de porcentaje el rendimiento en procesador y memoria RAM usada por cada equipo gestionado y una segunda herramienta llamada *Munin* que está instalada en cada una de las máquinas virtuales y de los host anfitriones donde se gráfica y se visualizan detalles más específicos del rendimiento, para poder tener otro parámetro de monitoreo adicional y así visualizar posibles fallas en el funcionamiento en los principales componentes de cada equipo individual.

Una vez hecho el monitoreo por el lapso de seis semanas se comprobó la estabilidad en cada uno de los servicios que están corriendo en producción (DNS caché, bactrack4 y servidor OWASP) y además se observó que cada componente virtual en almacenamiento y red no produjo errores al momento de transferir datos y/o archivos desde y hacia cada servidor virtual, produciéndose una ligera ralentización en la conexión remota SSH por las limitadas capacidades de cada servidor virtual.

Para mayor detalle el documento de plan e informe de pruebas ***[Anexo D]*** contiene los detalles correspondientes al análisis y los resultados que se dieron durante el proceso de pruebas.

CONCLUSIONES

- ✓ En la fase de análisis del presente proyecto se identificó la importancia de planificar adecuadamente; ya que se requirió realizar cuatro tipos de análisis, encuestas y reuniones con los administradores para identificar de manera más precisa los servicios a virtualizar; por lo que se utilizó la información de la infraestructura existente, de las plataformas de virtualización existentes, gestores de estas plataformas y los servicios que estaban activos e inactivos; valorando y estableciendo la criticidad de cada uno de estos para poder tener una referencia más exacta de las necesidades de simplificación de infraestructura y los beneficios que traerá el implementar cualquier solución de virtualización sobre todo del tipo Open Source.

- ✓ Del total de servicios operativos en la Universidad se logro identificar que ocho estaban clasificados como de menor criticidad, de los cuales por cuestiones de factibilidad y capacidades del servidor físico de pruebas; fueron implementados tan solo tres: servicio DNS caché y dos servicios de Auditoría de Seguridad tanto para redes como para aplicaciones web respectivamente justificados por su bajos niveles de carga y ocasional demanda.

- ✓ También se logró determinar en nuestras pruebas que la implementación futura de este tipo de solución es muy factible en servicios mucho más críticos que los implementados, para lo cual se necesita contar con la debida infraestructura IT y así minimizar la desventaja de ser un sistema con un posible punto central de fallo. Para mitigar lo anterior es fundamental contar con equipos físicos replicados, hardware redundante y configuraciones de almacenamiento lógico, permitiendo un buen nivel de balanceo de carga, contingencia a nivel de hardware y que el mismo soporte el cambio en caliente en software (live migration) y en hardware (hot-spare).

- ✓ La plataforma Xen, usada en este proyecto bajo entornos de producción logró una fiabilidad muy aceptable, logrando un rendimiento muy cercano al nativo, con una degradación de aproximadamente 8 al 15 por ciento por servicio, conforme al monitoreo realizado *(capitulo 4 sección D),* en nuestro entorno de pruebas bajo condiciones de ejecución regulares.

- ✓ El ahorro múltiple en costes en simplificación del equipos hardware, consumo de energía, licencias de software a mediano y largo plazo que implica el implementar esta tecnología es considerable que además permite colaborar con la reducción de emisión de gases de efecto invernadero con el que se han comprometido los países, sobre todo los desarrollados por lo que la tendencia sugiere un aumento progresivo de los llamados Centro de datos verdes o Data Green Centers.

- ✓ La solución Open Source planteada, si bien tiene muchas ventajas en ahorro de costes en cuanto a eliminar la subutilización y demanda de recursos de servidores, tiene una contra parte que es el soporte ante posibles bugs. Lo anterior nos hace recomendar que en las fases iniciales de este tipo de proyectos, se contrate dicho soporte en las herramientas de administración, debido a que algunos inconvenientes son detectables por usuarios con un nivel aceptable de expertise en la interpretación de logs, manejo de sistemas operativos, sistemas avanzados de almacenamiento y redes de datos.

- ✓ A lo largo de la investigación se asimiló conceptos totalmente nuevos como lo son la paravirtualización, los hipervisores, redes virtuales, routing, bridges y otros más, que me despertaron interés y permitieron confirmarla gran oportunidad de negocio que existe en el campo de la *Virtualización de Servicios y el Cloud Computing*; ya que por su ahorro en costes y retorno de inversión son tecnologías muy atractivas en la actualidad y con mucha proyección en los próximos años.

RECOMENDACIONES

- En etapas inexpertas de proyectos de virtualización se necesita de una buena planeación de la infraestructura del hardware así como de la red de datos, ya que para este tipo de proyectos es necesario tomar varias recomendaciones como el diseño adecuado del esquema de redes de comunicaciones; debido a que los servicios virtualizados a ejecutar, en gran parte están limitados por las características del hardware físico disponible por lo que se recomienda que en equipos limitados de recursos no se tenga elevadas tasas de generación de flujo de entrada y salida de datos.

- Es cierto el Open Source está en continuo auge, por lo que para proyectos de iniciación en Virtualización se recomienda el uso de herramientas que tengan simplicidad de uso y sobre todo un soporte continuo, es decir que se tenga asistencia para el mantenimiento, de manera que no sea tan traumático para el administrador el inmiscuirse paulatinamente en esta tecnología hasta que su nivel de expertise le permita una gestión y mantenimiento más avanzado.

- En la fases de planeación y desarrollo es necesario contar con todos los involucrados en el proyecto para que exista un consenso de los requisitos materiales, logísticos y humanos a usarse definiendo primeramente un entorno de pruebas adecuado, un control de cambios y principalmente tener el apoyo incondicional de la gerencia, ya que los costos iniciales de implantación, consolidación y/o migración a esta tecnología es costosa, pero totalmente justificable a corto y mediano plazo por los ahorros en los que se incurre.

- Para la fase de implementación es recomendable el uso de hardware que tengan características similares, ya que un punto importante es la homogeneidad del hardware porque contribuye a disminuir los problemas producidos por incompatibilidad entre plataformas de distinto procesador (Intel o AMD). Además se usan versiones de sistemas base de 64 bits porque aprovechan mucho mejor las capacidades del sistema hardware

del equipo anfitrión y permite un mayor aprovechamiento de los nuevos tipos de procesadores multi-núcleo y la ampliación de memoria para un mayor capacidad de alojamiento de host virtuales.

Bibliografía

1. Arias, D. (2009 Mayo). *Aplicaciones web.* From http://admonredes.wordpress.com/proyecto-2/
2. Barham, P., Dragovic, B., Fraser, K., Hand, S., Harris, T., Ho, A., et al. (2003). Xen and the Art of Virtualization. *Xen and the Art of Virtualization*.
3. Belmonte, A. A. (2007). Virtualización en GNU/Linux. *Virtualización en GNU/Linux*.
4. Buytaert, K. (2007). openQRM as Virtual Machine Manager. *openQRM as Virtual Machine Manager*.
5. Cayuqueo, S. (2007). *Comprendiendo Xen: un caso práctico de virtualización.* From http://www.whyfloss.com/pages/conference/static/editions/bsas07/charla10.pdf
6. Chaganti, P. (2007). *Xen Virtualization.* Packt Publishing.
7. Cherkasova, L., & Gardner, R. (2005). Measuring CPU overhead for I/O processing in the Xen virtual machine monitor. (pp. 24-24). USENIX Association.
8. Chisnall, D. (2007). *The Definitive Guide to the Xen Hypervisor.* Prentice Hall.
9. Christian, S. H. (2005). Xen 3.0 and the Art of Virtualization. *Proceedings of the Linux Symposium, II*, 320.
10. Clark, C. (2008). Xen User's Manual. *Xen User's Manual*.
11. Collado, E. (2010 йил febrero). *iSCSI que es y como funciona?* From http://eduangi.com/2010/02/09/iscsi-que-es-y-como-funciona/
12. Convirture. (2011). *Convirture*. From http://www.convirture.com
13. Corretgé, S. I. (2009). Introducción a la virtualización con Xen y KVM. *Introducción a la virtualización con Xen y KVM.*
14. devopsdays. (2009). *Cloud Computing Technology – Comparison Matrix.* From http://www.devopsdays.org/
15. Gartner. (2008 йил Abril). *http://www.gartner.com.* Retrieved 2010 йил Septiembre from http://www.gartner.com/it/page.jsp?id=638207
16. Gartner. (2010 йил Septiembre). *http://www.gartner.com*. Retrieved 2010 йил 14-Septiembre from http://www.gartner.com/it/page.jsp?id=1440213
17. Hagen, W. V. (2008). *Xen® Virtualization.* Wiley Publishing.
18. Hoopes, J. (2009). *Virtualization for security : including sandboxing, disaster recovery, high availability / John Hoopes.*
19. IBM. (2009). *Linux KVM - Kernel-based Virtual Machine - System x Virtualization.* From Linux KVM - Kernel-based Virtual Machine - System x Virtualization: http://www-01.ibm.com/redbooks/community/pages/viewpage.action?pageId=4718751

20. IDG. (2009 йил diciembre). *Pros y contras de la virtualización. Definición y soluciones de una tecnología emergente.* From http://www.idg.es/partnerzone/misioncritica/index.asp?seccion=articulos&id=199021
21. Jan, C. (2008). *Virtualization Guide 5.2.* Red Hat, Inc.
22. Jones, T. (2009). Anatomy of the libvirt virtualization library. *Anatomy of the libvirt virtualization library.*
23. Lash. (2007). *Harvard Business Review. Competitive advantage on a warming planet.*
24. López, H. (2009 йил Marzo). *RAID, LVM y ZFS.* From http://www.ldc.usb.ve/~lhector/raid-lvm-zfs.pdf
25. Matthews, J. (2008). *Running Xen: A Hands-On Guide to the Art of Virtualization.* Prentice Hall.
26. Menon, Aravind; Santos, Jose Renato; Turner, Yoshio; Janakiraman, G. (John); Zwaenepoel, Willy. (2005). Diagnosing performance overheads in the xen virtual machine environment. (pp. 13-23). ACM.
27. Microsoft Encarta. (2009). Enciclopedia Encarta. Efecto Invernadero. *Enciclopedia Encarta. Efecto Invernadero*.
28. Mingay, Simon. (2009). Green IT: The New Industry Shock Wave. *Green IT: The New Industry Shock Wave.*
29. Olcina, F. (2008 йил Julio). *Funcionamiento de Xen.* From Funcionamiento de Xen: http://www.arcos.inf.uc3m.es/~folcina/pfc-html/node24.html
30. OpenQRM. (2010). *OpenQRM*. From http://www.openqrm.com
31. Red Hat. (2008). *Linux Virtual Server (LVS) for Red Hat Enterprise Linux 5.2.* Red Hat, Inc.
32. Red hat. (2010). *Spacewalk*. From Spacewalk: http://spacewalk.redhat.com/
33. Red Hat Linux. (2010). *Virt-manager*. From Virt-manager: http://virt-manager.et.redhat.com/
34. Rule, D., & Dirtnner, R. (2007). *The Best Damn Server Virtualization Book Period.* (A. Williams, Ed.) Amorette Pedersen.
35. Sabater, P. J. (2006). Virtualización con Xen 3.0.3 en Debian Etch con kernel a medida para 32 y 64 bits. *Virtualización con Xen 3.0.3 en Debian Etch con kernel a medida para 32 y 64 bits.*
36. Shah, A. (2008). *Virtualización Profunda.* From http://www.linux-magazine.es/issue/36/034-036_KVMLM36.pdf
37. Smaldone, J. (2008). Virtualización de hardware. *Virtualización de hardware.*
38. Talens-Oliag, Sergio. (2009). Herramientas de Virtualización libres para sistemas GNU/LINUX.
39. tldp.org. (2009). *Instalación y configuración de Xen 3.0 en Debian.*

40. UTPL - Picoita, Galo. (2009). *Informe sobre la capacidad instalada a nivel de servidores en la UTPL.* Loja.
41. Vavai, M. (2010). *Las tecnologías de virtualización: VirtualBox y Xen.* From Las tecnologías de virtualización: VirtualBox y Xen: http://vavai.com/2010/01/25/teknologi-virtualisasi-virtualbox-xen-hypervisor-full-virtualization-paravirtualization/
42. VM Spain. (2010). *Hipervisor VMM.* From http://virtualization.com
43. Wiki XenSource. (2010). From http://wiki.xensource.com/xenwiki/XenNetworking
44. Wikipedia. (2009). *Wikipedia - VNC*. Retrieved 2010 йил Octubre from http://es.wikipedia.org/wiki/VNC
45. Wikipedia. (2010). Virtualización. *Virtualización*.
46. Williams, D. (2007). *Virtualization with Xen™: Including XenEnterprise™, XenServer™, and XenExpress™.* Syngress Publishing.
47. Woitasen, D. (2006). Virtualizando con Xen. Nov. 2006. *Virtualizando con Xen. Nov. 2006.*

ANEXOS

Anexo A. Resultados de encuesta

Resultados de encuesta aplicada a los administradores de servicios de la UTPL.

Anexo B. Nomenclatura – Información y estadística de servidores.

Nomenclatura – Lista equipos IBM-Blade

ITEM	Servidor	Administrador	Dependencia	Modelo Server	Tipo Procesador	Procesador			Memoria		Disco Interno				Disco Externo			Fecha de Producción	Proyección en años
						Num.	Velocidad GHZ	Uso %	Tamaño GB	Uso %	Num.	Capacidad GB	Uso %	Uso %	Num.	Capacidad GB	Uso %		
IBM1	INTRANETCITTES	Ana Lucia Abad	Dir. G CITTES	X SERIES 225	Intel (R) Xeon(TM) CPU 2.80GHz	2	2.8	80 %	2	50 %	1	30 GB	70%	70%				02/01/2005	3
IBM2	GDR1BACK.UTPL.EDU.EC	Diana Alexandra Torres Guarnizo	Gestión del Conocimiento	IBM x Series 345	Intel(R) Xeon(TM) CPU 2.40GHz	2	3.6	10 %	3	15 %	2	2 discos 70 GB	100%	100%				07/04/2003	5
IBM3	CAJANUMA	Juan Carlos Morocho	Grupo de Desarrollo de Software	pSeries 520 IBM,8203-E4A	PoerPC Power6 2 procesadores dual core de 4.2 GHz	9	4.2	99 %	10	99 %	4	584GB	94.8%	95%	4	1.2TB	30%	06/04/2009	4
IBM4	CALSERVER	Viviana Montaño	Grupo de Desarrollo de Software	IBM-System x3650	Dual-core Xeon	2	3	10 %	4	10 %	1	68,2 GB	30%	30%				02/04/2007	5
IBM5	NTS01	Danilo Jaramillo H	Grupo de Desarrollo de Software	xSeries 220	intel pentium III	2	1.26	5%	0.7	40 %	2	16 Gb	60%	60%				01/01/2001	10
IBM6	ASTERISK	Daniela Calva	Grupo de Telecomunicaciones	XSeries 226	Intel(R) Xeon(TM) CPU 3.20GHz	1	3.2	2%	1	40 %	2	75	57%	57%				23/08/2006	5
IBM7	ASTERISK-BACKUP	Daniela Calva	Grupo de Telecomunicaciones	System x3400	GenuineIntel	2	1.5	10 %	1	75 %	1	75 GB	25%	25%				01/01/2008	5
IBM8	GDR2	Ruperto Alexander López Lapo	Grupo de Telecomunicaciones	IBM x Series 305	Intel(R) Pentium(R) 4 CPU 3.06GHz	1	3.06	6%	1	60 %	1	36G	40%	40%				01/01/2007	5
IBM9	GDR5	Ruperto Alexander López Lapo	Grupo de Telecomunicaciones	xSeries 220	intel pentium III	2	1.23	5%	1	65 %	2	80G, 36G	40%	40%				01/01/2007	5
IBM10	GDR7	Ruperto Alexander López Lapo	Grupo de Telecomunicaciones	IBM Netfinity 3000	Pentium II (Deschutes)	1	0.44	15 %	0.31	90 %	2	9 G, 36G	52%	52%				02/01/2007	3
IBM11	MONITOREO	Carlos Aguilar M.	Grupo de Telecomunicaciones	IBM x Series 345	Intel(R) Xeon(TM) CPU 2.40GHz	1	3.06	4%	1	65 %	1	80GB	50.22 %	50.22 %				01/01/2006	3
IBM12	NOCAT	Daniela Calva	Grupo de Telecomunicaciones	Netfinity 3500 M20	GenuineIntel Pentium III 733 MHz	1	0.72	3%	0.73	89 %	1	40GB	85%	85%				05/05/2009	1
IBM13	OSSIM	Julia Pineda	Grupo de Telecomunicaciones	IBM-System x3650	Dual-core Xeon	3	3	30 %	4	78 %	2	140 Gb	60%	60%				01/11/2008	5
IBM14	REPO	Ruperto Alexander López Lapo	Grupo de Telecomunicaciones	IBM x Series 345	Intel(R) Xeon(TM) CPU 2.40GHz	3	2.4	1%	2	14 %	3	290GB	78%	78%				04/10/2006	5
IBM15	VIEWER	Julia Pineda	Grupo de Telecomunicaciones	xSeries 220	intel pentium III	1	1.26	25 %	1	21 %	1	17Gb	93%	93%				27/06/2007	1
IBM16	WEBMAIL	Ruperto Alexander López Lapo	Grupo de Telecomunicaciones	IBM xSeries 346	Genuine Intel, Intel(R) Xeon(TM) CPU 3.60GHz	2	3.6	44 %	2	60 %	2	146.8 GB	90%	90%				05/05/2007	5
IBM17	SERVERHOSPITAL	Jenny Maria Cuenca	Hospital UTPL	XSeries 226	Intel(R) Xeon(TM) CPU 3.20GHz	2	3.2	30 %	2	90 %	2	70GB	50%	50%	4	700GB	10%	28/06/2007	3
IBM18	F-SECURE	Janeth Alba	Soporte Técnico	x Series 235	Intel(R) Xeon(TM)	1	2.8	1%	2	87 %	2	160 GB	49%	49%				20/03/2007	5
IBM19	ULOJA	Juan Pablo Ureña Torres	Soporte Técnico	pSeries 550 - IBM,9133-55A	PowerPC_POWER5 2-core 2.1 GHz	2	2.1	90 %	18	99 %	4	584GB	75%	75%	4	584GB	90%	11/10/2007	5

Anexo C. Configuración de los servicios virtualizados.

Configuración del servicio DNS Caching.

Los paquetes necesarios para implementar el servicio DNS Caché son:

bind	Este es el servidor DNS, servicio llamado named.
bind-utils	Utilerías complementarias para realizar consultas DNS.
bind-libs	Librerías usadas por los dos programas previos.
bind-chroot	Crea un subdirectorio especial donde se "enjaula" (chroot) bind, esto con objetivos de brindar seguridad.
caching-nameserver	Archivos de configuración para un servidor DNS Cache.

Para realizar lo anterior modificamos el archivo de configuración "named.conf", ubicado en /etc.

Hacemos una copia del archivo original, en caso de necesitar en un futuro el archivo original:

```
# cp named.conf named.conf.bak
```

/var/named/chroot/etc

Luego, el siguiente paso es modificar el archivo de configuración de red que permite indicarle al sistema cual es nuestro DNS master. Actualiza el archivo "/etc/resolv.conf" en donde comentamos las líneas correspondientes a los DNS para que se presente lo siguiente:

```
# vi /etc/resolv.conf


#nameserver 172.16.50.58
#nameserver 172.16.50.55
```

```
nameserver 127.0.0.1
```

Con esto hecho, indicamos que es nuestro propio servidor DNS Cache, el que resuelve las consultas de los usuarios cuando accedan a un sitio. Tan solo comentamos las previamente establecidas y agregamos la dirección localhost 127.0.0.1 como nameserver. Luego digitamos el comando:

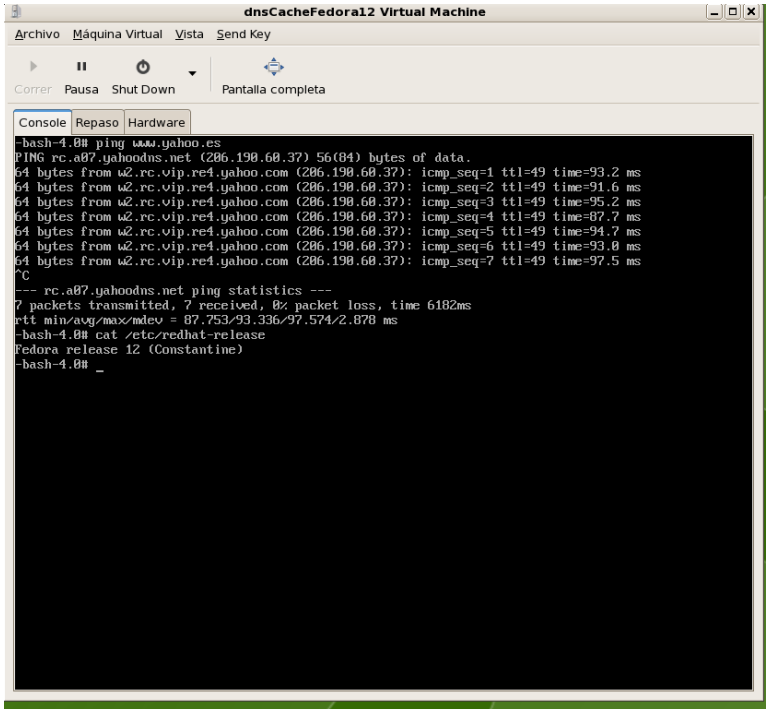
```
# service named start
Iniciando named:   [ OK ]
```

Con esto hemos iniciado el servicio exitosamente. Ahora para evitar que nuestro archivo de configuración /etc/resolv.conf se altere cuando se obtenga dirección por dhcp, se agrega la siguiente línea al final de las que ya existen o simplemente verifica que ya exista y modifícala si fuera necesario. Con esto se logra que no se modifique el archivo ***/etc/resolv.conf***.

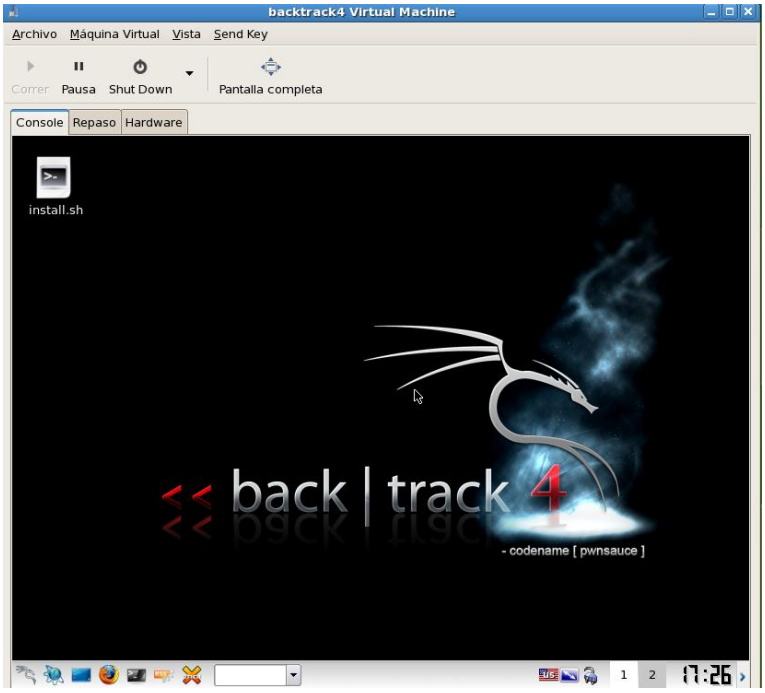
PEERDNS=no

Finalizamos aplicando las reglas de firewall a nuestro servicio.

iptables -A INPUT -s 172.16.0.0/24 -p udp --dport 53 -j ACCEPT



Configuración del servicio de auditoría Backtrack 4.



El archive de configuración del equipo virtual con sus elementos se detalla a continuación:

name = "backtrack4"
uuid = "a6dacda7-014c-ef01-d16f-285b0999a152"
maxmem = 1024
memory = 512

```
vcpus = 2
builder = "hvm"
kernel = "/usr/lib/xen/boot/hvmloader"
boot = "c"
pae = 1
acpi = 1
apic = 1
localtime = 0
on_poweroff = "destroy"
on_reboot = "restart"
on_crash = "restart"
device_model = "/usr/lib64/xen/bin/qemu-dm"
sdl = 0
vnc = 1
vncunused = 1
keymap = "en-us"
disk = [ "file:/images/bt4.img,hda,w", ",hdc:cdrom,r" ]
vif = [ "mac=00:16:36:5e:07:e0,bridge=xenbr1,script=vif-
bridge,ip=172.16.189.55,vifname=bt4" ]
parallel = "none"
serial = "pty"
```

Configuración del servicio de auditoría OWASP.

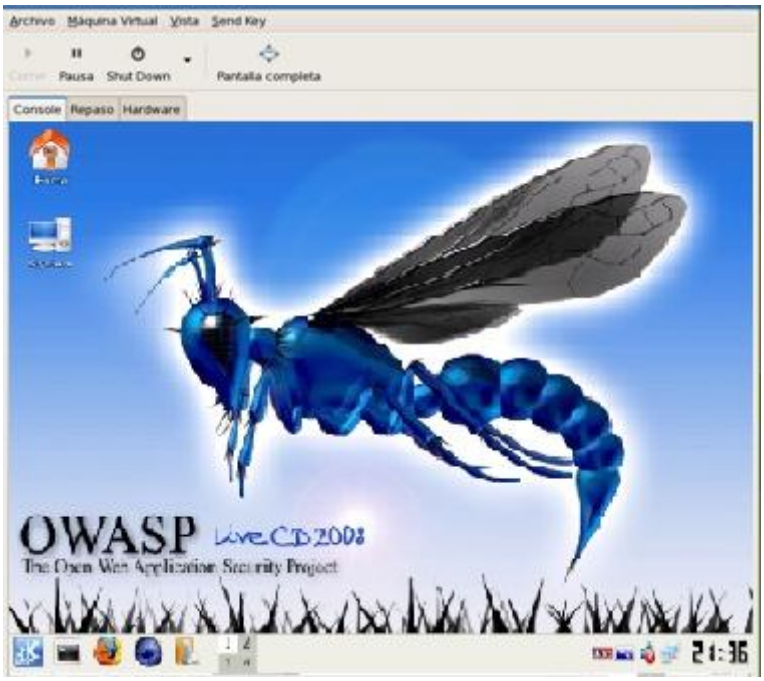


Figura 36. Pantalla de inicio de la distro OWASP.

Figura 37. Interfaz y menú de herramientas de OWASP.

Los archivos de cada equipo virtual creado se almacenan en el directorio /etc/xen.

A continuación presentamos el archivo de configuración con sus respectivos parámetros y componentes básicos del equipo virtual de auditoría llamado audit_oswap:

```
# xen domU config file
name = "audit_owasp"
memory = "512"
disk = [ 'phy:/dev/dsk/vowasp.img,xvda,w' ]
vif = [ 'mac=00:16:3e:13:e4:81, ip=172.16.189.55,bridge=xenbr0' ]
uuid = "5aafecf1-dd66-401d-69cc-151c1cb8ac9e"
bootloader="/usr/bin/pygrub"
vcpus=2
vnc=1
on_reboot = 'restart'
on_crash = 'restart'
```

Anexo D. Plan de Pruebas

Plan de Pruebas
Proyecto – Virtualización de Servicios no críticos con Plataformas Open Source

Versión [1.0.0]

Información del documento

TÍTULO:	Plan de pruebas inicial
SUBTÍTULO:	Proyecto - Virtualización de servicios no críticos de la UTPL con plataformas Open Source.
VERSIÓN:	[1.0.0]
ARCHIVO:	PlanPruebasVirtualizacion.doc
AUTOR:	Juan Carlos Ordóñez G.
ESTADO:	Inicial

Lista de cambios

VERSIÓN	FECHA	AUTOR	DESCRIPCIÓN
1.0.0	2010-08-16	JCOG	Emisión Inicial

Firmas y aprobaciones

ELABORADO POR:	Juan Carlos Ordóñez		
FECHA:	2010-08-16	Firma:	
REVISADO POR:	Ing. Alexander López Ing. Diana Torres		
FECHA:	2010-08-17	Firma:	
APROBADO POR:	Ing. Alexander López Ing. Diana Torres.		
FECHA:		Firma:	

Plan de Pruebas

Introducción

Propósito

El plan de pruebas tiene por objeto, verificar que cada equipo virtual y servicios del proyecto funcionen correctamente ante escenarios representativos del entorno en el que se ejecutará y validar que el proyecto desarrollado, corresponda a los requisitos expresados inicialmente.

En este documento se reúne la información necesaria, para planear y controlar el desarrollo de las pruebas de verificación y validación de un entorno del laboratorio de virtualización de servicios.

El documento plan de pruebas tiene los siguientes objetivos:

- Identificar la información del proyecto: documentación generada en la etapa de análisis y los componentes que deben ser probados.
- Describir las estrategias de prueba a ser empleadas.
- Identificar los recursos necesarios y suministro estimado de los esfuerzos de pruebas.
- Listar los productos entregables del proyecto de pruebas.

Alcance

Las pruebas se enfocarán a la implementación de un laboratorio de virtualización de servicios utilizando plataformas Open Source, que incluye los siguientes módulos:

- Configuración.
- Logística.
- Reportes.

Además se enfocarán hacia la funcionalidad de los productos obtenidos a los cuales se incluirán los siguientes niveles de pruebas:

- Pruebas unitarias (componentes virtuales/ unidades simples).
- Pruebas de aceptación (realizadas por el usuario final), para verificar la confiabilidad del servicio antes de ser puesto oficialmente en marcha.

Para cada una de las fases indicadas anteriormente, existirán los siguientes Tipos de Prueba:

- Pruebas de funcionalidad de cada componente configurado en la máquina virtual. (discos virtuales, tarjetas de red, asignación de procesadores y RAM.)
- Pruebas de I/O de red (conectividad - ssh, http, icmp -, respuesta en milisegundos, paquetes recibidos y perdidos).
- Pruebas de rendimiento de cada máquina anfitrión u host generado por las VMs.(Monitoreo NOC)

Audiencia

La audiencia involucrada en el proyecto es la siguiente:

- Implementador (Juan Carlos Ordóñez).
- Usuarios (Ing. Julia Pineda, Ing. Alexander López)

Objetivo y factores que motivan las pruebas

Garantizar que el servicio final este implementado aceptablemente, que satisfaga las necesidades, expectativas y objetivos planteados inicialmente en el proyecto; en relación a la definición, ejecución y control de los procesos que se llevan a cabo en la implementación.

Misión

Los principales objetivos de la iteración plan de pruebas son:

- Encontrar fallas de forma eficiente.
- Encontrar problemas significativos.
- Validar documentación generada.
- Evaluar y mitigar riesgos percibidos en cuanto a la funcionalidad de cada componente virtual.
- Brindar satisfacción a los usuarios o administradores de cada servicio.
- Recomendar mejoras en las actividades al plan de pruebas.
-

Factores de motivación

Es necesario efectuar pruebas por varias razones entre ellas:

- Evaluar y mitigar riesgos del proyecto.
- Evaluar y mitigar riesgos técnicos.
- Detectar posibles fallas o defectos.

Estrategia de pruebas

La estrategia de pruebas presenta la aproximación recomendada para las pruebas de acuerdo al objetivo que persigue. Para cada tipo de prueba, se deberá suministrar una descripción de la misma y por qué está siendo implementada y ejecutada.

Las principales consideraciones para la estrategia de pruebas son las técnicas a ser usadas y el criterio de conocimiento cuando la prueba este completa.

Referente a la documentación, conforme se van utilizando las diferentes herramientas se va generando la información relacionada.

Tipos de pruebas

Pruebas de integridad de servidores virtuales.

Las pruebas de integridad de VMs, buscan comprobar que el acceso y manipulación de los componentes de los equipos virtuales sean los correctos y que sus resultados están de acuerdo a los datos de prueba utilizados.

Objetivo de la Prueba:	**Asegurar que el acceso a cada equipo virtual funciona correctamente.**
Técnica:	Invocar cada método y proceso de acceso al equipo virtual, con datos válidos de configuración. Inspeccionar los archivos de configuración para asegurar que los componentes virtuales y sus parámetros han sido cargados como se pretendía.

Criterio de Conclusión:	Todos los métodos de acceso, procesos y componentes de la VM funcionan correctamente y sin inconsistencias en los archivos de configuración.

Pruebas de aceptación del usuario

Las pruebas de aceptación están diseñadas para asegurar al cliente (en este caso el administrador), que se construyó el equipo virtual estipulado; se caracterizan por tener al cliente como testigo y se ejecutan en la plataforma que van a operar.

Objetivo de la prueba:	**Asegurar al cliente que cada equipo virtual satisface sus necesidades planteadas.**
Técnica:	Se ejecutan todas las opciones del servicio para verificar: · Que los equipos virtuales son fáciles de usar. · Que se realizan todas las funciones especificadas por el cliente.
Criterio de Conclusión:	Que las pruebas planeadas se ejecuten satisfactoriamente. Todos los defectos identificados han sido corregidos.

Metodología

La metodología que guiará el desarrollo de las pruebas antes descritas, comprende:

- **Casos de prueba:** Se construirá en base a la especificación detallando la siguiente información:
  - Número de caso de prueba.
  - La funcionalidad del componente sobre el que se realiza la prueba.
  - Código de caso de prueba.
  - Descripción del caso de prueba.
  - Condición o datos de entrada.
  - Datos de Entrada
  - Salida esperada
- **Bitácora de errores**: Se generará con los resultados de la ejecución de los casos de prueba, ésta contendrá la siguiente información:

- Número de error.
- Nombre de probador que detecta el error
- Fecha en que fue registrado el error
- Elemento de Prueba
- Tipo de Error
- Prioridad
- Estado
- Fecha de cierre
- Nombre del error
- Descripción del error.
- Encargado de la corrección del error.

Los errores detectados en el desarrollo de los casos de prueba, se han tipificado como:

- ***Defecto:*** Aquellos que no permiten continuar con la ejecución del servidor virtual, o su presencia producirá resultados incorrectos en la ejecución de otras funcionalidades relacionadas al servicio.
- ***Incidente:*** Son errores que no detienen el funcionamiento del sistema; sin embargo, producen funcionalidad ligeramente incorrecta, como por ejemplo al tratar de montar una imagen .iso en una unidad CD-ROM y el resultado obtenido no es el esperado.
- ***Discrepancia:*** Son situaciones que pueden ser mejoradas, no afectan en el funcionamiento del sistema.

Los estados que se han definido para los errores son:

- ***Abierto:*** Significa que no se ha podido tomar medidas correctivas para el error (bug) y que este continúa presente.
- ***Cerrado:*** En este estado se encuentran los errores que han sido debidamente corregidos y los casos de prueba que los generan se han ejecutado nuevamente, para verificarlos.

- **Informe de resultados de pruebas:** Este producto contendrá como su nombre lo indica, los resultados de las pruebas ejecutadas.

Recursos

Esta sección presenta los roles recomendados para la disciplina de pruebas en el proyecto actual. Cada recurso presenta sus responsabilidades, conocimientos y habilidades.

Recursos humanos

Esta tabla presenta los recursos que participarán en las actividades de pruebas.

Recursos Humanos		
Recurso	**Cantidad recomendada**	**Responsabilidades específicas**
Administrador/Diseñador de pruebas	1	Identifica, prioriza e implementa casos de pruebas. Provee coordinación general de actividades de pruebas. Responsabilidades: · Apoyo a generar plan de pruebas. · Diseñar los casos de pruebas.
Probador	1	Ejecuta las pruebas. Responsabilidades: · Ejecutar las pruebas. · Registrar resultados. · Documentar errores.
Administración de equipos virtuales	1	Asegura que los equipos virtuales están preparados para las pruebas.

Recursos tecnológicos

A continuación se detallan los recursos tecnológicos (hardware) del proyecto bajo pruebas.

Recursos del Sistema (Hardware de Pruebas)	
Recurso	**Nombre/Tipo**
Servidor	vxen01 /Virtualización
—Red/SubNet	Local -172.16.30.0/24
—Nombre de Servidor	vxen01
—Nombre de servicios	dnsCache

Recursos del Sistema (Hardware)	
Recurso	**Nombre/Tipo**
Servidor	vxenaudit /Virtualización
—Red/SubNet	Local -172.16.189.0/24
—Nombre de Servidor	vxenaudit
—Nombre de servicios	Auditoria - Backtrack4 y Auditoría OWASP.

Responsabilidades

EQUIPO	FUNCIONES	COMPONENTES
Juan Carlos Ordóñez	Adm. Pruebas Diseñador de casos de pruebas.	Aplicación y testing.
Juan Carlos Ordóñez	Probador	Documentación de análisis y diseño del proyecto.
Ing. Julia Pineda	Administrador(a) servidores de Auditoría.	Administrador/Evaluador del equipo virtual.

Entregables

Informe de pruebas

El informe de pruebas, contendrá los resultados de la aplicación de los casos de prueba, además incluirá:

- Casos de pruebas.
- Lista de Errores.
- Informe de resultados pruebas.

Riesgos

Se describen los riesgos que pueden presentarse al desarrollarse las pruebas.

- Retraso en la entrega del servidor y/o servicio virtual.
- Tiempo elevado en la corrección de los defectos encontrados.
- Ausencia de quienes validan las pruebas, lo cual puede provocar retrasos.

Glosario

DR	Documento de Requerimientos.
ANSI	American National Standards Institute
IEEE	The Institute of Electrical and Electronics Engineers, Inc.
VM	Virtual Machine.
CP	Caso de prueba.

Informe de resultados de pruebas

Proyecto Virtualización de servicios menos críticos con plataformas Open Source.

Versión [1.0.0]

Información del documento

TÍTULO:	Informe de resultado de pruebas
SUBTÍTULO:	Virtualización de servicios no críticos con plataformas Open Source
VERSIÓN:	[1.0.0]
ARCHIVO:	InformePruebas.doc
AUTOR:	Juan Carlos Ordóñez
ESTADO:	Borrador

Lista de cambios

VERSIÓN	FECHA	AUTOR	DESCRIPCIÓN
1.0.0	2010-11-06	JCOG	Emisión Final

Firmas y aprobaciones

ELABORADO POR:	Juan Carlos Ordóñez		
FECHA:	2010-11-06	Firma:	
REVISADO POR:	Ing. Diana Torres G.		
FECHA:		Firma:	
APROBADO POR:	Ing. Diana Torres G.		
FECHA:		Firma:	

Informe de resultados de pruebas

En base al documento denominado plan de pruebas, se realizaron las pruebas de: unidad (de cada máquina virtual) y aceptación del usuario, utilizando las estrategias definidas para el desarrollo de cada una de ellas.

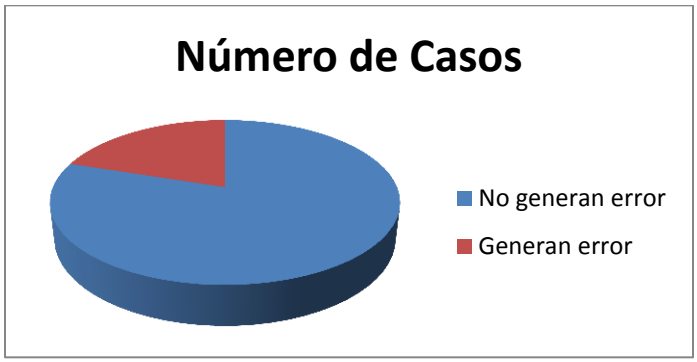
La ejecución de las pruebas mencionadas, nos permitió verificar que cada componente de la plataforma funciona correctamente ante escenarios representativos del entorno en el que se ejecutará y validar que el software y plataformas implementadas, corresponden a los requisitos y objetivos expresados en el inicio del proyecto.

- *Porcentaje de casos de prueba que generaron error.*

CASOS DE PRUEBA	Nro.	%
No generan error	8	80
Generan error	2	20
TOTAL:	**10**	**100**

Se encontró que el 20% de los casos de prueba definidos, para los distintos módulos de la plataforma, produjeron errores de diferentes tipos (configuración y acceso) y con distintas prioridades; cabe indicar que la totalidad de estos errores fueron corregidos. Frente a esta situación tenemos que el 80% de los casos de prueba ejecutados, no produjeron errores.

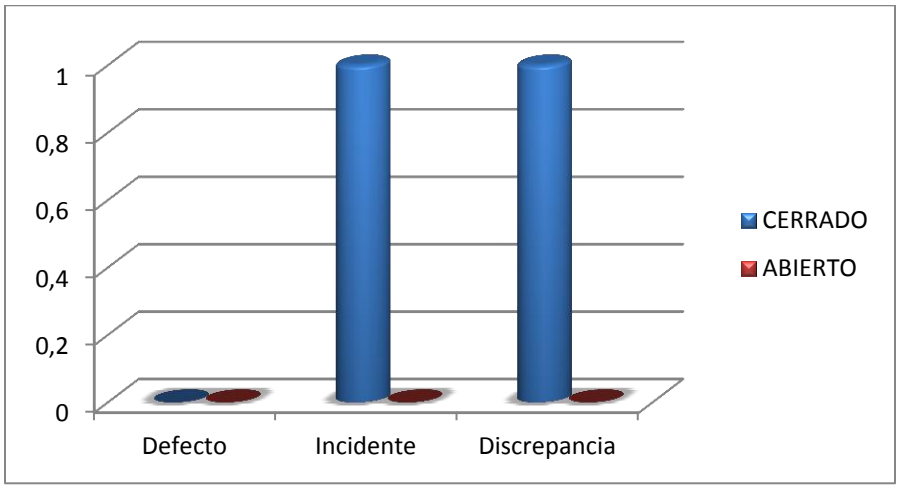
En la figura que sigue, se muestra la relación entre el número de casos de prueba definidos, frente al número de casos de prueba que generaron error.



- *Incidencia de los tipos de errores encontrados y su estado actual.*

TIPO DE ERROR	ESTADO		TOTAL	%
	CERRADO	ABIERTO		
Defecto	0	0	0	0
Incidente	1	0	0	50
Discrepancia	1	0	1	50
TOTAL	**2**	**0**	**2**	**100**

La Figura que sigue abajo, representa los diferentes tipos de error encontrados en el desarrollo de las pruebas, respecto del estado en el que éstos se encuentran.



El número de errores que se dieron en los casos de prueba corresponden al 50% al tipo incidente y al 50% tipo discrepancia por ser errores de permisos de

acceso a nivel de red y configuración respectivamente.

Cabe destacar, que la totalidad de los errores encontrados han sido corregidos, por lo cual se encuentran en estado cerrado.

- *Funcionalidades del sistema que presentaron errores en la ejecución de los casos de prueba correspondientes.*

FUNCIONALIDAD	NRO.ERRORES	%
No se pudo configurar el bridge en interfaz principal.	1	50
No se pudo acceder vía ssh al servidor virtual.	1	50
TOTAL	**2**	**100**

Pruebas de aceptación del usuario

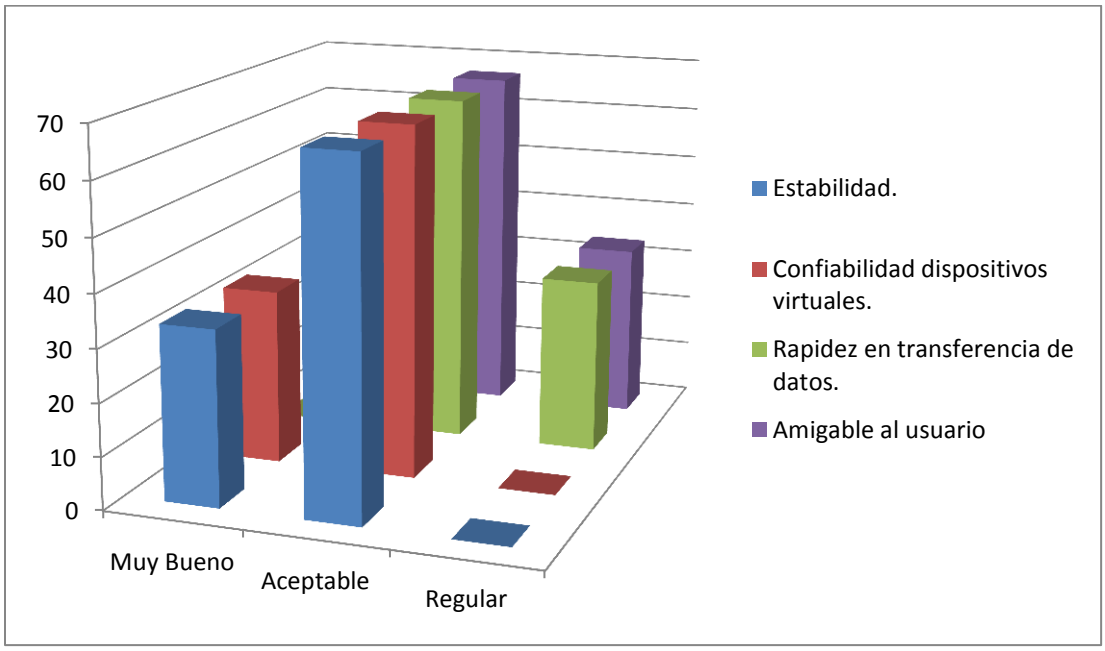
La aplicación de estas pruebas permitió conocer el nivel de aceptación de los usuarios que serán los encargados del manejo y administración de la plataforma y de los hosts de virtualización; para lo cual, luego de una explicación básica del funcionamiento del mismo, hemos realizado una capacitación a los ahora administradores y usuarios de los hosts virtuales.

Lista de Encuestados

NOMBRE DEL USUARIO	CARGO
Ing. Julia Pineda	Administrador servicios
Juan Carlos Ordóñez	Administrador servicios
Cesar Montalván	Gestor Investigador – Usuario servicios

Los resultados de las pruebas se detallan a continuación:

	Comprobación de estabilidad.	Confiabilidad en los dispositivos virtuales.	Rapidez en la transferencia de datos.	Fácil de usar y amigable al usuario
Muy Bueno	33,3	33,3	0	0
Aceptable	66,7	66,7	66,7	66,7
Regular	0	0	33,3	33,3
TOTAL	**100%**	**100%**	**100%**	**100%**



Conclusiones:

- De la figura anterior, podemos concluir que: para el 66,7% de los usuarios consideran que la estabilidad y la confiabilidad de dispositivos virtuales es muy buena y otro 33,3% lo considera aceptable. En cuanto a la transferencia de datos un 33% considera que tiene un muy buen tiempo de respuesta, mientras que el 66,7% considera que la confiabilidad de los datos es aceptable, esto se explica por las características del hardware de red, ya que por una sola interfaz pasa todo el tráfico de todos los servicios y se ralentiza y finalmente tenemos un 33,3% que la considera como aceptable al manejo de la aplicación y un 66,7% piensa que es regular esto debido a que se necesita

de un tiempo de capacitación familiarización con las herramientas ya que los usuarios aun no están muy involucrados con nuestro proyecto.

a) Modo Puente (bridge)

Esta configuración de puentes de red dentro de dom0 permite que todos los domUs aparezcan en la red como hosts independientes. Este método lo que hace es tomar una IP dentro de la red donde se conecta, ya sea IP estática o dinámica y simplemente permitir a sus máquinas virtuales el utilizar una o varias tarjetas Ethernet virtuales para unirse a una o varias redes existentes.

Para poder tener esta configuración solamente tenemos que descomentar las siguientes líneas del archivo de configuración de XEN.

```
(network-script network-bridge)
(vif-script vif-bridge)
```

Luego en el archivo de configuración de nuestro equipo virtual revisamos que tenga agregada línea:

```
vif = [ "mac=00:16:3e:56:16:d6,bridge=xenbr0" ]
```

Para aplicar los cambios reiniciamos el proceso de Xen llamado xend.

```
#service xend restart
```

Bridge para varias interfaces Ethernet.

En Xen podemos utilizar varias tarjetas de red, de manera que tendríamos creadas en nuestro host anfitrión interfaces numeradas con la nomenclatura xenbr# y peth#. Lo anterior puede ayudar en el tráfico de las DomUs ya que podemos distribuir de mejor manera el tráfico de nuestras interfaces virtuales por distintas rutas de red. Para esto creamos un siguiente script y lo guardamos en la ruta de scripts en /etc/xen/scripts/.

```
#vim multi-bridge.sh
        #!/bin/sh
        dir=$(dirname "$0")
        "$dir/network-bridge" "$@" vifnum=0 netdev=eth0 bridge=xenbr0
        "$dir/network-bridge" "$@" vifnum=1 netdev=eth1 bridge=xenbr1
```

Hemos creado dos puentes o bridges dentro de XEN. Ahora solo tenemos que modificar el archivo de configuración de XEN.

vim /etc/xend/xend-config.sxp

Y aquí comentamos la línea:

#(network-script network-bridge)

Para luego agregar la línea de nuestro script creado debajo de la comentada:

(network-script multi-bridge.sh)

Configurado Xen con dos puentes, reiniciamos el servicio xend para aplicar las modificaciones.

/etc/init/xend restart

b) Modo Encaminador o router.

Esta configuración se aplica cuando:

- Cuando la máquinas están en una diferente LAN.
- Cuando el tráfico se red dirige hacia el exterior por la red 192.168.1.0/24
- Son máquinas visibles desde 192.168.1.0/24

Es necesario modificar el archivo de configuración de Xen. Cometamos las líneas correspondientes a las configuraciones de NAT y Bridge y descomentamos la de network-route y vif-route.

```
(network-script network-route)
(vif-script vif-route)
```

Comentar las siguientes:

```
#(network-script network-bridge)
#(network-script network-nat)
#(vif-script vif-bridge)
#(vif-script vif-nat)
```

Una vez habilitado el nodo route, el próximo paso es configurar las interfaces virtuales de los domUs, ahora tendremos que modificar archivo de un domU, en nuestro caso va hacer oswap. En esta línea del archivo del equipo virtual oswap tiene la configuración de red en bridge.

vif = ["mac=00:16:3e:56:16:d6,bridge=xenbr0"]

Tenemos que modificarla o mejor comentarla y crear otra línea debajo de esta con los siguientes datos

vif = ["ip=10.0.0.2"]

Ahora tenemos que habilitar el modo route siguiente forma:

```
echo 1 > /proc/sys/net/ipv4/conf/all/proxy_arp
iptables -t nat -A POSTROUTING -s 10.0.0.0 -j MASQUERADE
route add -net 10.0.0.0 netmask 255.255.255.0 gw 192.168.1.5
```

Con esto ya tenemos configurada la red en modo route, tan solo reiniciamos XEN.

c) Modo NAT

Esto configuración aplica cuando:

- DomU están en una LAN privada.
- DomU hacen NAT con Dom0 para poder llegar a la otra LAN y parece como si el tráfico procede de Dom0.

- Para poder habilitar esta configuración tenemos que editar el archivo de configuración de XEN.

- Buscar las siguientes líneas y descomentarlas:

(network-script network-nat)

(vif-script vif-nat)

Comentamos las siguientes líneas de las otras configuraciones:

```
#(network-script network-bridge)

#(network-script network-route)

#(vif-script vif-bridge)

#(vif-script vif-route)
```

Hay que modificar la configuración de la MV.

vif = ["mac=00:16:3e:56:16:d6,bridge=xenbr0"]

Tenemos que modificarla o mejor comentarla y crear otra línea debajo de esta con los datos del ip correspondiente.

vif = ["ip=10.0.0.2"]

Luego de realizar esta modificación, tan solo nos resta reiniciar XEN para que se apliquen los cambios.

Anexo F. Archivos de configuración de XEN.

Archivo principal de configuración Xen en /etc/xen/xend-config.spx

Aquí mostramos las partes usadas del archivo de configuración, y que nos permiten lograr una funcionalidad deseada y acorde a nuestras preferencias.

```
# -*- sh -*-
#
#La línea que sigue nos permite habilitar (yes) o deshabilitar (no) la conexión vía http.
(xend-http-server yes)

#Aquí habilitamos o deshabilitamos el acceso vía línea de comandos para la administración del servidor.
(xend-unix-server yes)

Las líneas que siguen son para habilitar otros formatos de comunicación como es XML a través de llamada a procedimiento remoto o RPC.
(xend-tcp-xmlrpc-server yes)
(xend-unix-xmlrpc-server yes)

Para habilitar la posibilidad migración entre servidores físicos se necesita habilitar esta línea.
(xend-relocation-server yes)

(xend-unix-path /var/lib/xend/xend-socket)

# Cuando se habilita el servidor http es necesario establecer esta línea para escucha del puerto.
(xend-port            8000)

# En el caso de estar habilitada la migración es necesario establecer el puerto de escucha.
(xend-relocation-port 8002)

# Para el acceso desde otra locación o dirección IP que no sea el servidor propio se descomenta #esta línea. Aquí se puede especificar la notación usando una IP o un rango de direcciones de las #que se puede acceder a los servidores.
(xend-address '172.16.0.0/24')
#(xend-address localhost)

# Aquí especificamos los servidores a los que se puede migrar las máquinas virtuales. Aquí se permite la reubicación a cualquier servidor pero por seguridad se recomienda especificar los host a los que se permite esta acción.
(xend-relocation-address '172.16.28.170')

# Aquí se especifica las direcciones IP o nombres de servidores que van a estar autorizados
#a que se les permita administrar la migración entre hosts.
(xend-relocation-hosts-allow '172.16.28.170')
```

```
#(xend-relocation-hosts-allow '^localhost$ ^localhost\\.localdomain$')


# Luego tenemos la configuración de la red en modo bridge, esta vez especificando la interfaz
# eth1, por lo tanto el nombre del bridge en este caso será “xenbr1”.
(network-script 'network-bridge netdev=eth1')

(vif-script vif-bridge)

# Como observamos las otras opciones de configuración de red han sido deshabilitadas.
#(network-script network-route)
#(vif-script     vif-route)

#(network-script network-nat)
#(vif-script     vif-nat)

# Mínimo de memoria aceptable para el funcionamiento de dom0.
dom0-min-mem 256)

# El control de acceso por conexión remota VNC.
(vnc-listen '172.16.28.229') //IP-rango IP de la máquina de administración

(vncpasswd 'xxxxxxxx') //especificamos un password para acceso vía VNC

# Idioma de teclado en el anfitrión.
(keymap 'en-us') //por defecto.
```

Anexo G. Manual de Instalación de Convirture 2.0.1

Instalación del Convirt Management Server (CMS) en la distribución Linux Ubuntu 10.04.

Primeramente debemos asegurarnos de contar con todos los privilegios de red necesarios ya que es necesario descargar paquetes y sus dependencias respectivas. En el caso de contar con un servidor proxy entonces establecemos y exportamos la variable en el entorno especificando el nombre ip del servidor y el número de puerto correspondiente:

- ***export http_proxy="172.16.50.54:3128"***

Luego nos logueamos con el usuario que se va a encargar de la administración. Es preferible que no se autentique como root o superusuario por razones de seguridad en privilegios.

Ahora descargamos el paquete wget.

- ***sudo apt-get install wget***

Descargamos los paquetes de Convirture 2.

- ***wget - no-cache http://www.convirture.com/downloads/convirt/2.0.1/convirt-install-2.0.1.tar.gz***
- ***wget - no-cache http://www.convirture.com / downloads/convirt/2.0.1/convirt-2.0.1.tar.gz***
- ***wget - no-cache http://www.convirture.com/downloads/convirture-tools/2.0.1/convirture-tools-2.0. 1.tar.gz***

Descomprimimos el directorio principal para crear el entorno TurboGears necesario para que corra la aplicación web de gestión de Convirt.

- ***tar-xzf convirt-install-2.0.1.tar.gz***

Dentro de la carpeta descomprimida nos ubicamos en la siguiente ruta para empezar con la instalación de las dependencias necesarias para iniciar la instalación:

- ***sudo ./convirt-install/install/cms/scripts/install_dependencies***

NOTA: En Ubuntu (Lucid10.04) se necesitan paquetes adicionales y se tiene que ejecutar:

- ***sudo apt-get instalar libmysqlclient-dev python2.6-dev python-setuptool***

Además cuando la instalación le pida tendrá que asignar una contraseña al root de la base de datos MySQL, entonces se teclea convirt. Esto luego es modificable.

Se necesitan configurar algunos parámetros para la base de datos. Para esto en la sección del archivo de configuración de MySQL (/etc/mysql/my.cnf) bajo la sección [mysqld] agregamos las siguientes lineas:

- ***innodb_buffer_pool_size=1G***
- ***innodb_additional_mem_pool_size=20M***

Posterior reiniciamos el servicio MySQL:

- ***/etc/init.d/mysqld restart***

Por defecto Convirt se instala en el directorio home del usuario ***~/convirt***. Si se desea instalar en otra ubicación se debe editar la variable CONVIRT_BASE en archivo que está en la ruta del directorio descomprimido anteriormente en ***install/cms/scripts/install_config.***

Por ejemplo, cuando cambiamos de:

CONVIRT_BASE=~ a CONVIRT_BASE=~/cms

El CMS ahora será instalado en la ruta ~/cms/convirt.

Ahora descomprimimos el instalador del CMS:

- ***tar -xzf ./convirt-2.0.1.tar.gz -C $CONVIRT_BASE***

Prosiguiendo instalamos el entorno de TurboGears.

- ***./convirt-install/install/cms/scripts/setup_tg2***

Modificamos el archivo sqlalchemy.url en la ruta src/convirt/web/convirt/development.ini para compaginar usuario y password para la base de datos MySQL.

- ***sqlalchemy.url=mysql://root:convirt@localhost:3306/convirt?charset=utf8***

Y luego ejecutamos :

- ***./convirt-install/install/cms/scripts/setup_convirt***

NOTA: Por seguridad cada vez que el proceso se inicie se le preguntará por un password/passphrase para asegurar la identidad SSH sobre el servidor CMS.

Una vez ingresado el password/passphrase ya se puede levantar el proceso de convirt. Para esto ingresamos al directorio y ejecutamos el script convirt-ctl de la siguiente manera:

- ***cd ~/convirt***
- ***./convirt-ctl star***

Para que la conexión no sea interrumpida por el firewall es necesario aplicar la regla:

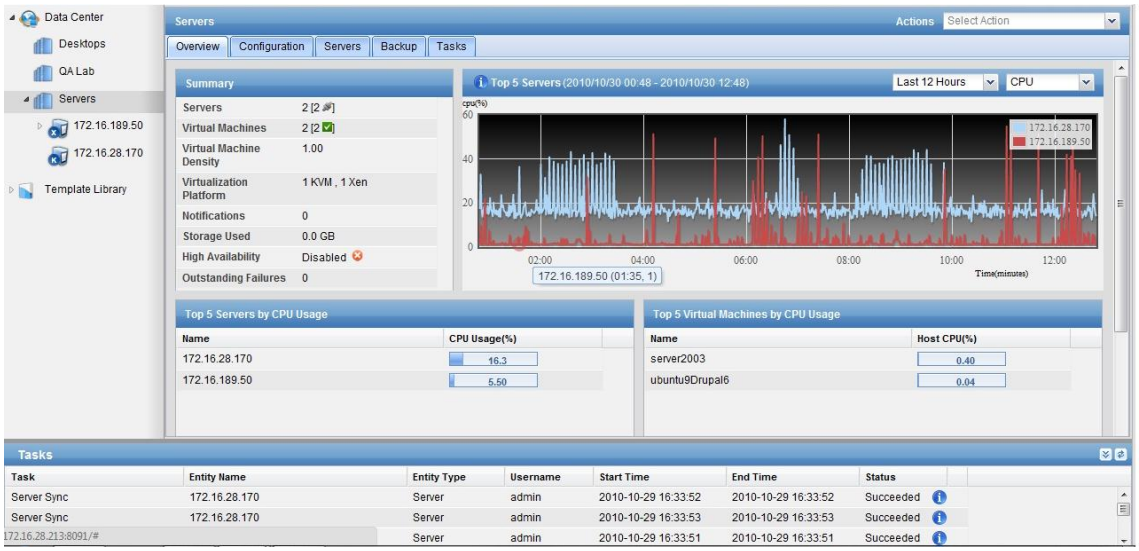
- ***iptables -I INPUT -p tcp --dport 8081 -j ACCEPT***

Luego abrimos un navegador para autenticarnos en el CMS :

En nuestro caso nuestro CMS está corriendo en la URL:

http://172.16.28.229:8081/.

Las credenciales por defecto son admin/admin, pero es importante cambiarlas tras el primer ingreso por seguridad.



Para detener el servidor de administración usamos los comandos de stop en el script convirt-ctl:

- ***cd ~/convirt***
- ***./convirt-ctl stop***

- **Instalación del parche convirt-tools en el server anfitrión.**

La secuencia de comandos convirt-tools le ayuda a hacer los cambios y configuraciones necesarias en el servidor host para ser administrado y manejado por ConVirt (CMS).

Para la plataforma de Xen, la ejecución de este comando configura el servidor xend para escuchar en el puerto 8006 y abre el puerto 8002 para la migración. El comando también detecta el puente (bridge) por defecto y escribe un resumen de sus operaciones sobre el directorio /var/cache/convirt/server_info.

Los pasos son los siguientes:

Ingresar al servidor CMS con su cuenta de usuario CMS.

Copia el comprimido de convirt-tools (Ejm: scp) en el servidor host a gestionar y luego lo descomprimimos.

- ***tar -xzf convirture-tools-2.0.1.tar.gz***

Para leer la ayuda tecleamos:

- ***./convirt-tool -h***

Para comprobar la plataforma antes de hacer cualquier cambio:

- ***./convirt-tool --detect_only setup***

Posteriormente instalamos las dependencias:

- ***./convirt-tool install_dependencies***

Para conectar cada equipo virtual a la red se necesita de bridges. Si su configuración no lo tiene se crearan por cada interfaz física de red que tenga. Para más información verifique su modo de red primero verifique con el comando brctl show.

- ***./convirt-tool setup***

Este script de configuración puede desconectar la red momentáneamente.

Si tiene configurado y funcionando la red en modo puente (bridge) y/o quiere gestionar esto manualmente se puede saltar esta configuración:

- ***./convirt-tool --skip_bridge setup***

Para la versión Xen 4.0 /SLES 11/SLES 11 SP1, xen server puede configurarse con la opción SSL:

- ***./convirt-tool --xen_ssl --all setup***

En Ubuntu se debe comprobar que los siguientes puertos estén abiertos:

Por tanto: el puerto de ssh (por lo general 22)

Para Xen: el puerto TCP 8002 para permitir la migración, 8006 para permitir ConVirt interactuar con el servidor Xend.

Para KVM: los puertos TCP son del 8002 a 8012 para la migración.

Para verificar que los puertos están en escucha usamos el siguiente comando sobre los servidores gestionados:

- ***netstat –nlp | grep 80***