Universidad del Aconcagua

Facultad de Ciencias Sociales y Administrativas

Licenciatura en Telecomunicaciones

Autor: Palmieri, Emilio Nicolás.

Tutor: Ing. Guillermo Sández.

Legajo: 18621.

Título: Comparación entre sistemas Virtualizados y Paravirtualizados.

Lugar: Mendoza – Argentina.

Fecha: 02 de septiembre de 2015.

CALIFICACIÓN:

ÍNDICE

Resumen Técnico.

La presente tesina tiene como objetivo comparar las tecnologías de Virtualización y Paravirtualización, a fin de tomarlas y representar los pasos metodológicos y lineamientos para su aplicación. Con esto se pretende asistir al usuario a la hora de definir la tecnología a aplicar y tener en claro cuál es la adecuada para su emprendimiento o proyecto. Se describen ambos métodos detalladamente para comprender sus funcionamientos y características, y se realiza una implementación de cada una de estas tecnologías, a fin de poder realizar una comparativa entre ellas que tenga en cuenta las problemáticas que se observan a nivel de implementación. Para elaborar el piloto de Virtualización, se utiliza el software VMware Workstation 10. Y la Paravirtualización se lleva a cabo con el sistema de Citrix Xen-Server. Para ambos casos, se instalan dos servidores con las aplicaciones correspondientes, y sobre ellos se realiza un set de pruebas y ensayos de funcionamiento y rendimiento, de modo de poder extraer sus diferencias y decidir cual de estos es el que se adapta mejor a las necesidades de la empresa. Finalmente se presentan las conclusiones obtenidas tras la realización de las pruebas y ensayos realizados, además de la propia experiencia adquirida durante la implementación del proyecto, presentando así cuál es el método más adecuado para la necesidad de la empresa según sus requerimientos y sus disponibilidades.

Capítulo 1. Introducción.

1.1 Definición del Problema.

El problema surge de observar el crecimiento de necesidades informáticas que tienen las empresas en la actualidad, donde cada vez se solicitan más servicios, tanto para proporcionar a los clientes como a los propios empleados. El avance tecnológico actual, insta a la empresa a adoptar servicios como: plataformas Web, Servidores de Datos, Telefonía IP, etc.

Al momento de proporcionar todos estos servicios dentro de la Empresa, se debe disponer de equipamiento hardware y software de base, y espacio para su instalación, lo que multiplica los costos de inversión, de operación y mantenimiento.

Para resolver esta problemática, se ha desarrollado de manera amplia y cada vez con mayor grado de adopción, las tecnologías de Virtualización.

La necesidad de recurrir a estas tecnologías motiva la necesidad de crear una “guía” para tener en claro los requisitos necesarios y las consideraciones básicas para la elección del método de Virtualización que se adapte mejor a las necesidades de la Empresa.

1.2 Alcance.

La presente Tesina desarrolla los conceptos fundamentales de las tecnologías de Virtualización y Paravirtualización, y realiza una detallada comparación entre estas dos tecnologías.

Para ello se instalará una solución para cada una de estas tecnologías y se virtualizarán los mismos servidores. Esto se realizará con el mismo equipamiento para poder obtener resultados comparativos válidos de pruebas a ambas tecnologías.

Los sistemas virtualizados serán Tirxbox que es una distribución del sistema operativo GNU/Linux, basado en CentOS, que tiene la particularidad de ser una central telefónica (PBX) por software basada en la PBX de código abierto Asterisk. Como cualquier central PBX, permite

interconectar teléfonos internos de una compañía y conectarlos a la red telefónica convencional (RTB - Red telefónica básica).

El otro sistema que se virtualizará es Windows Server 2003 que es un sistema operativo de la familia Windows de la marca Microsoft para servidores que salió al mercado en el año 2003. El cual proporciona servicios de DNS, DHCP, Servidor de archivos, de impresoras, de aplicaciones, etc.

Una vez instalados estos sistemas se realizan pruebas de Benchmarking (técnica utilizada para medir el rendimiento de un sistema o componente del mismo, frecuentemente en comparación con el que se refiere específicamente a la acción de ejecutar un benchmark). De los cuales se obtendrán las comparaciones de rendimiento de cada sistema de vitualización, y sus recomendaciones de uso.

Dentro del alcance de la presente tesina se incorporan también las experiencias y recomendaciones propias para los procesos de implementación y desarrollo de tecnologías de virtualización de Empresas.

1.3 Objetivos:

1.3.1 Principal:

Realizar una comparación detallada entre las tecnologías de Virtualización y Paravirtualización, a partir de pruebas a una instalación piloto de cada uno de los sistemas.

1.3.2 Secundarios:

- Clarificar los conceptos que distinguen a las tecnologías de virtualización y paravirtualización.
- Implementar un piloto con cada una de las tecnologías.
- Dar recomendaciones y experiencias propias que faciliten la implementación de este tipo de tecnologías.

- Recomendar la selección de cada una de estas tecnologías en función del entorno de negocios en la que se implemente.

1.4 Viabilidad:

Dado que existen hoy cada vez mas empresas que adoptan tecnologías de Virtualización y Paravirtualización para implementar las múltiples aplicaciones informáticas que utilizan, resulta sumamente necesario tener un conjunto de parámetros que le permitan a éstas seleccionar la mejor solución para cada caso de negocio.

La tesina será una herramienta de suma utilidad para aquellos profesionales que necesiten tomar una decisión de implementación de este tipo de tecnologías.

Capítulo 2. Marco Teórico.

2.1 Virtualización.

En informática, el término virtualización se refiere a la abstracción de los recursos de una computadora, este es llamado Hypervisor o Monitor de Maquina Virtual (Virtual Machine Monitor – VMM) que crea una capa de abstracción entre el hardware de la máquina física (equipo host-anfitrión) y el sistema operativo de la máquina virtual (equipo guest-invitado).

Esta capa de software (VMM) es la que se encarga de manejar, gestionar y arbitrar los cuatro recursos principales de una computadora (Memoria, CPU, Almacenamiento y Red) y así poder repartir dinámicamente dichos recursos entre todas las máquinas virtuales definidas en el equipo anfitrión, permitiéndonos tener varios sistemas virtuales ejecutándose sobre el mismo equipo físico.

La virtualización a su vez crea una interfaz externa que permite esconder una implementación subyacente, ya sea mediante la combinación de recursos en localizaciones físicas diferentes, o a través de la simplificación del sistema de control. En los últimos años, el desarrollo de nuevas plataformas así como de nuevas tecnologías de virtualización ha hecho que el concepto de virtualización sea una práctica común en distintos entornos empresariales. *[ROS, 2008]*

2.1.1 Historia de la Virtualización.

A sus inicios (en 1959), la virtualización era conocida como “time sharing” o tiempo compartido, gracias al profesor Christopher Strachey, un profesor de la Universidad de Oxford y líder del Grupo de Investigación en Programación. A Strachey, esta técnica le permitía escribir el código fuente de un programa mientras otro programador compilaba otro programa. *[WILLIAMS, 2007]*

Dos años más tarde, se desarrollo uno de los primeros sistemas operativos de tiempo compartido, el CTSS (“Compatible Time-Sharing System”). El CTSS es considerado el abuelo de los sistemas operativos de tiempo compartido ya que influye en el desarrollo, entre otros, de:

- IBM M44/44X.
- MULTICS, que influye fuertemente en la familia UNIX (Linux).
- CP/M, que influye fuertemente en 86-DOS, el cual deriva en Microsoft Windows.

Durante el año 1962, La Universidad de Manchester desarrollo una de las primeras supercomputadoras mundiales, ésta aprovecho conceptos como el tiempo compartido, la multiprogramación, y el control compartido de periféricos. Fue apodada “Atlas Computer”. Un proyecto dirigido por el Departamento de Ingeniería Eléctrica y financiado por Ferranti Limited, la Atlas fue el equipo más rápido de su tiempo. Su velocidad de procesamiento se debió en parte a una separación de los procesos del sistema operativo en un componente llamado el supervisor y el componente responsable de la ejecución de los programas de usuario. El supervisor gestionaba recursos claves, como el tiempo de procesamiento de la computadora, y pasaba instrucciones especiales o extra-códigos, para ayudar a administrar y gestionar el entorno de las instrucciones del programa de usuario. En esencia, este fue el nacimiento del “hipervisor” o monitor de máquina virtual (VMM).

Además, la Atlas introdujo el concepto de memoria virtual, llamada One-Level Store, y técnicas de paginación para la memoria del sistema. Este almacenamiento también se separó lógicamente del almacenamiento utilizado por los programas de usuario, aunque los dos se integraron. En muchos sentidos, éste fue el primer paso hacia la creación de una capa de abstracción que todas las tecnologías de virtualización tienen en común. En la figura 1 se observa el Laboratorio donde se encontraba instalada la Computadora Altas. *[WILLIAMS, 2007]*

Figura 1: Computadora Atlas, Universidad de Manchester, Diciembre de 1962.
Fuente: http://www.dte.eis.uva.es/Docencia/ETSII/SMP/BAK/Ha_SComp/historia.htm

Durante 1965, IBM también incursionó con la virtualización con el Proyecto M44/44X. Creado en el centro de IBM Thomas J. Watson Research en Yorktown, Nueva York, el proyecto creó una arquitectura similar a la del Atlas. Esta arquitectura fue la primera en utilizar el término “máquinas virtuales” y se convirtió en la contribución de IBM a los conceptos emergentes de sistemas de tiempo-compartido. La máquina principal era un IBM 7044 (M44) y varias máquinas virtuales 7044 simuladas (las 44Xs), usando hardware, software, paginación, memoria virtual y multiprogramación, respectivamente.

En el año de 1964 fue el Centro Científico de Cambridge de IBM quienes desarrollaron el CP-40, un sistema operativo que implementa la herramienta virtualización completa (full-virtualization), permitiendo simular 14 “pseudos-máquinas”, más tarde llamadas máquinas virtuales.

Para 1966, el Centro Científico de Cambridge de IBM empieza la conversión del CP-40 y el CMS para ejecutarlos en el S/360-67. El CP-67 es una significativa re implementación del CP-40 y es la primera implementación ampliamente disponible de la arquitectura de “máquina virtual”.

Durante 1968, se da a conocer National CSS (NCSS), una compañía que explora la idea de ofrecer servicios de tiempo compartido, aprovecha la disponibilidad de CP/CMS para iniciar la implementación de VP/CSS ya que el rendimiento de CP/CMS no es rentable para sus planes de comerciales.

Al llegar los años 70, IBM empieza a desarrollar 'CP-370/CMS', una completa re implantación del 'CP-67/CMS' para su nueva serie 'System/370' (S/370).

Posteriormente, IBM anuncia el primer sistema operativo de máquina virtual de la familia VM (VM/CMS), el 'VM/370' (basado en 'CP-370/CMS') y destinado para 'System/370' con hardware de memoria virtual. El 'VM/370' se basa en dos componentes; CP (Control Program) y CMS (Conversational Monitor System). La función más importante del nuevo CP es la habilidad de ejecutar una VM dentro de otra VM. Todo esto fue desarrollado durante 1972. La familia de relaciones y procedencias de CP/CMS se observa en la figura 2. *[WILLIAMS, 2007]*

<table>
<tr><th colspan="3">Relaciones de la Familia CP/CMS</th></tr>
<tr><td colspan="3">→ Derivación >> Fuerte Influencia > Influencia/Procedencia</td></tr>
<tr><td rowspan="5">CTSS</td><td colspan="2">> IBM M44/44X</td></tr>
<tr><td>>> CP-40/CMS → CP[-67]/CMS</td><td>→ VM/370 → VM/XA versions → VM/ESA → z/VM
→ VP/CSS</td></tr>
<tr><td colspan="2">> TSS/360</td></tr>
<tr><td colspan="2">> TSO for OS/MVT → for OS/VS2 → for MVS → ... → for z/OS</td></tr>
<tr><td colspan="2">>> MULTICS and most other time-sharing platforms</td></tr>
</table>

Figura 2: Árbol de la Familia CP/CMS.
Fuente: en.wikipedia.org/wiki/History_of_CP/CMS.

Paralelamente la National CSS (NCSS), aporta con el VP/CSS a la serie 'System/370'. El VP/CSS mejora el rendimiento del CSS utilizando paravirtualización, a través de llamadas directas al hypervisor con la instrucción no virtualizada DIAG, en lugar de simular las operaciones de bajo nivel de los comandos de E/S.

Ya para la década de los 80 y con la llegada de las relativamente económicas máquinas x86, comenzó una nueva era de micro computadoras, aplicaciones cliente-servidor; en donde los enormes y potentes “mainframes” con mil y una tareas y utilidades en una sola caja gigantesca se

comenzaron a cambiar por relativamente pequeños servidores y computadoras personales de arquitectura x86, lo que se convirtió rápidamente en el estándar de la industria.

Debido a esto, una vez más, el tema de la virtualización vuelve a quedar prácticamente en el olvido, y no es hasta finales de la década de los 90 que gracias al alto desarrollo del hardware volvemos a caer en un predicamento similar al que estábamos en los años 60: el hardware existente es altamente eficiente, y utilizar cada equipo para una sola aplicación seria un desperdicio de recursos, espacio, energía y dinero; y tampoco es conveniente asignarle múltiples usos o instalar varias aplicaciones en un solo servidor convencional, por más de una razón. Es por esto que vuelve a resurgir la idea de dividir el hardware, de manera tal que funcione como múltiples servidores independientes pero compartiendo los recursos de un mismo servidor físico. Y es de aquí que nace lo que hoy todos conocemos como “Virtualización”.

Actualmente existen diferentes compañías que se dedican al desarrollo de aplicaciones y soluciones de virtualización. Para la década de los noventa, la empresa VMware inventó la virtualización para la plataforma x86 para abordar los problemas de infrautilización y de otras índoles, a lo largo de un proceso que obligó a superar gran cantidad de desafíos. *[VMware, 2007]*

Una vez conocida el hecho histórico de esta tecnología se aprecia el hecho de que la virtualización empezó a implementarse de la mano de IBM, como una manera lógica de particionar ordenadores mainframe en máquinas virtuales independientes. Estas particiones permitían a los mainframes realizar múltiples tareas: ejecutar varias aplicaciones y procesos al mismo tiempo. Dado que en aquella época los mainframes eran recursos caros, se diseñaron para particionar como un método de aprovechar al máximo la inversión.

Así que con todo lo que la historia dejo detrás de nosotros, y con tantas empresas que afirman usar el término de virtualización, la definición que surge para ésta es:

“La Virtualización es, un marco o metodología que divide los recursos de un equipo informático en varios entornos de ejecución, mediante la aplicación de uno o más conceptos o tecnologías, como el hardware y el software de particionamiento, tiempo compartido, la

simulación parcial o total de la máquina, la emulación, la calidad del servicio, y muchos otros." *[WILLIAMS, 2007]*

Así como lo hizo durante la década de 1960 y principios de 1970 con VM/370 de IBM, la virtualización permite que múltiples instancias modernas del sistema operativo se ejecuten simultáneamente en un solo equipo, aunque mucho menos costosos que los mainframes de aquellos días. Cada instancia de S.O. comparte los recursos disponibles en el hardware físico común, tal como se ilustra en la figura 3.

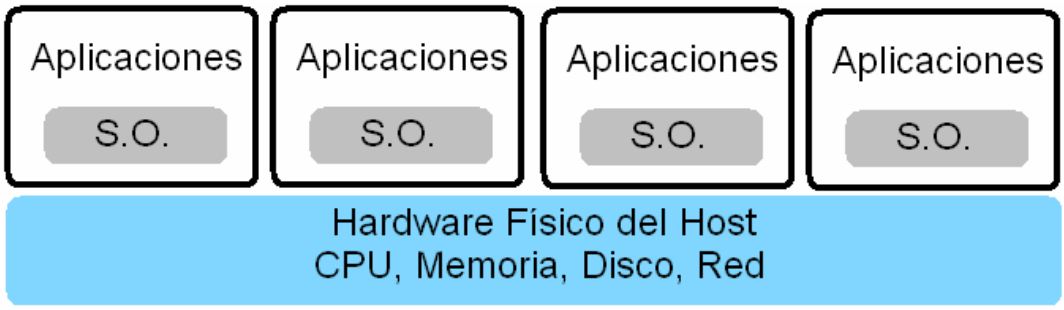

Figura 3: Las M.V. montadas sobre el hardware físico.

Fuente: Virtualization with Xen, Pág. 9.

2.1.2 Conceptos.

Aunque este tema se ha ampliado gradualmente en estos últimos años, el término Virtualización se ha convertido en uno de los conceptos más usados últimamente entre la comunidad de usuarios domésticos que quieren disponer de varios sistema operativos en un solo ordenador. Básicamente, virtualización es una tecnología que permite instalar y configurar múltiples computadoras y/o servidores completamente independientes (conocidas como máquinas virtuales) en un solo equipo, ya sea una computadora o servidor.

A pesar de que estas máquinas virtuales comparten todos los recursos de un mismo hardware, cada uno trabaja de manera totalmente independiente (con su propio sistema operativo, aplicaciones, configuraciones, etc.). Por ejemplo; en lugar de utilizar 5 servidores físicos, cada uno de ellos corriendo una aplicación que solo utiliza el 10% de los recursos de su servidor; podemos

instalar 5 máquinas virtuales, cada una con su propia aplicación y configuraciones específicas, en un solo servidor y utilizar el 50-60% de los recursos del mismo. *[WILLIAMS, 2007]*

Cabe señalar que cada una de estas máquinas virtuales, después de haber sido configuradas correctamente, deberán funcionar exactamente igual que un servidor o PC física (con conexión a una red, ingreso a un dominio, aplicar políticas de seguridad, conexión remota, etc.).

La manera en cómo funciona esta tecnología, es mediante el llamado Hypervisor o VMM (Virtual Machine Monitor), que crea una capa de abstracción entre el hardware de la máquina física (también llamado Host - Anfitrión), y el Sistema operativo de la máquina virtual (también llamado Guest - Invitado). Esta capa de software (VMM) maneja, gestiona, y arbitra los cuatro recursos principales de una computadora: CPU (Unidad Central de Procesos), Memoria, Red y Almacenamiento; y repartiendo dinámicamente dichos recursos entre todas las máquinas virtuales definidas en el computador principal. De esta manera podemos tener varios ordenadores virtuales ejecutándose sobre el mismo ordenador físico.

La virtualización crea una interfaz externa que esconde una implementación subyacente mediante la combinación de recursos en localizaciones físicas diferentes, o por medio de la simplificación del sistema de control. Un avanzado desarrollo de nuevas plataformas y tecnologías de virtualización han hecho que se vuelva a prestar atención a este importante concepto.

La principal ventaja de la virtualización es que permite tener varios ordenadores virtuales funcionando simultáneamente bajo un mismo hardware, y eso se consigue gracias a que un sistema operativo actúa como anfitrión, dotando parte de sus recursos a un sistema invitado, que es básicamente, el sistema operativo que se va a virtualizar.

Finalmente obtendremos una implementación que contará con las siguientes ventajas:

- Más económica: Requiere menos hardware, menos electricidad, menos enfriamiento, menos espacio, menos infraestructura, y menos tiempo de administración. Todo esto al final se traduce en ahorro de dinero.
- Menos compleja: Por las mismas razones mencionadas en el punto anterior, ya que existen menos cantidades de elementos pero se cumple con la misma demanda.

- Consume menos energía y espacio: Ya que ayuda a la protección del medio ambiente ahorrando energía y espacio.
- Más segura: Con los niveles de seguridad adecuados, una red virtual cuenta con menos puntos de ataque físicos, lo que la hace más segura. La virtualización es una excelente estrategia de seguridad al momento de elaborar un plan de backup o recovery.
- Más fácil de administrar: Con el conocimiento de virtualización y dejando de lado el "temor al cambio", administrar una red virtual debe ser más sencillo que administrar una red regular.

2.1.3 Tipos de Virtualización.

La virtualización se puede hacer desde un sistema operativo Windows o Linux, sea Windows XP, Ubuntu, o cualquier otra versión que sea compatible con el programa que utilicemos, en el que virtualizamos otro sistema operativo como Linux o viceversa, que tengamos instalado Linux y queramos virtualizar una versión de Windows. Al momento de virtualizar disponemos de los siguientes tipos. *[WILLIAMS, 2007]*

2.1.3.1 Virtualización por Hardware

Esta Virtualización asistida por Hardware funciona con extensiones introducidas en la arquitectura de procesador x86 para facilitar las tareas de virtualización al software corriendo sobre el sistema.

En computadores con arquitectura de x86, se cuenta con cuatro niveles de privilegio o "anillos" de ejecución, desde el cero (de mayor privilegio), que se destina a las operaciones del kernel del S.O., al tres (con privilegios menores) que es el utilizado por los procesos de usuario, al momento de utilizar este tipo de virtualización, se introduce un anillo interior o Anillo-1 que será el que un Hypervisor o VMM (Monitor de Maquina Virtual) que se usará para aislar todas las capas superiores de software de las operaciones de virtualización.

2.1.3.2 Virtualización de Almacenamiento.

Se refiere al proceso de abstraer el almacenamiento lógico del almacenamiento físico, y es comúnmente usado en SANs (Storage Area Network - Red de Área de Almacenamiento). Los recursos de almacenamiento físicos son agregados al "storage pool" (almacén de almacenamiento), del cual es creado el almacenamiento lógico.

2.1.3.3 Particionamiento.

Es la división de un solo recurso (casi siempre grande), como en espacio de disco o ancho de banda de la red, en un número más pequeño y con recursos del mismo tipo que son más fáciles de utilizar. Esto es muchas veces llamado “zoning”, especialmente en almacenamiento de red.

2.1.3.4 Máquina Virtual.

Se refiere básicamente como un sistema de virtualización, denominado "virtualización de servidores", que dependiendo de la función que esta deba de desempeñar en la organización, todas ellas dependen del hardware y dispositivos físicos, pero casi siempre trabajan como modelos totalmente independientes de este. Cada una de ellas con sus propias CPUs virtuales, tarjetas de red, discos etc. Lo cual podría especificarse como una compartición de recursos locales físicos entre varios dispositivos virtuales.

2.1.4 Grupos de Virtualización.

Dentro de los tipos de virtualización, debemos encerrar tres grupos muy importantes en esta tecnología como son:

2.1.4.1 Emulación.

La emulación de hardware simula cada instrucción del procesador como si de otro hardware se tratara.

En este tipo de virtualización, la máquina virtual (MV) es empleada para emular únicamente, un determinado tipo de hardware. De ésta manera, cada instrucción debe ser simulada por el hardware subyacente, la principal referencia es que es un sistema lento con respecto a los otros tipos de virtualización. Aun así, este tipo de virtualización está muy extendida entre los desarrolladores de firmware para hardware que todavía no ha sido fabricado, o que está en fase experimental. Dentro de los emuladores de software conocidos tenemos el Bochs y el Qemu.

2.1.4.2 Bochs.

Es un sistema de emulación bajo licencia LGPL (software libre) y que es capaz de simular diferentes arquitecturas, todo ello mediante el hardware subyacente. De ésta manera, se puede simular un ordenador x86 que es portable entre varias plataformas: x86, PowerPC, Alpha, SPARC y MIPS. Bochs es un sistema de virtualización en el que se simula el ordenador completo: procesador, periféricos, tarjetas gráficas, adaptadores de red. Permite la creación (emulación) de cualquier arquitectura, y la ejecución de múltiples sistemas operativos sobre Linux (Windows 95/98/NT/2000, FreeBSD, OpenBSD, etc).

Es un sistema muy poco reconocido principalmente porque carece de una interfaz gráfica muy amigable y sencilla, como ofrecen el resto de herramientas de virtualización.

2.1.4.3 Qemu.

Es una opción de virtualización que ofrece dos modos de operación:

- Modo de emulación de sistema completo, y
- Modo de emulación de usuario.

El primer modo es similar a Bochs ya que emula un ordenador completo (procesador, sistemas de almacenamiento y periféricos) y puede emular diferentes tipos de arquitecturas(x86, x86-64, etc). De esta forma se puede emular Windows y Linux, sobre un sistema Linux, Solaris y FreeBSD. El segundo modo solamente se puede alojar en Linux, y nos permite ejecutar binarios para arquitecturas MIPS, ARM, SPARC, PowerPC, u otras en desarrollo, sobre un Linux instalado en una arquitectura x86.

Al igual que Bochs, Qemu tampoco tiene una interfaz muy amigable con lo que su uso no está muy extendido. Hay que rescatar algo muy importante, Qemu fue la base sobre la que luego se desarrollaron otras tecnologías de virtualización, como entre otras, Xen.

2.1.4.4 Virtualización Completa.

La virtualización completa es aquella reconocida en gran parte de los ambientes virtualizados de la actualidad. Este tipo de tecnología, envía las instrucciones de la MV (Máquina Virtual) al procesador físico.

La virtualización completa es una técnica mucho más extendida que la técnica de la Emulación, ya que permite la posibilidad de ejecutar un sistema operativo sobre otro totalmente distinto instalado en la máquina física.

La técnica empleada para ello se basa en una máquina virtual (o Hypervisor VMM) que media entre el sistema operativo instalado en la Maquina Virtual y el hardware físico. El Hypervisor se sitúa entre el hardware real y el sistema operativo virtual ofreciendo con ello la posibilidad de ejecutar un sistema operativo tal cual. Es el encargado de realizar las traducciones pertinentes de las instrucciones máquina (generadas por la máquina virtual) para que puedan ser interpretadas por el procesador físico. El único requisito es que dicho SO esté diseñado para ese hardware en concreto.

Este tipo de virtualización obtiene mejores rendimientos si el procesador da soporte a instrucciones virtuales, como es el caso de las tecnologías VT y PACIFICA de Intel y AMD respectivamente. Esto es debido a que el procesador es capaz, en cierta forma, de interpretar las instrucciones generadas por la MV, sin ser necesaria su traducción. Cabe acotar que será esta la

tecnología implementada en el proyecto mediante un procesador AMD con tecnologías V de fábrica. Esta tecnología es una de las más reconocidas y estandarizadas por sus amigables interfaces con el usuario y la facilidad de configuración, entre estas está: VMware.

2.1.4.5 Paravirtualización.

Además de la virtualización completa y la emulación, también existe otra técnica de virtualización conocida como Paravirtualización. Éste, es un sistema virtualizador mediante el cual, las instrucciones de la VM se ejecutan directamente en el procesador físico, puesto que emplea sistemas operativos modificados para ello.

La Paravirtualización es una variante de la virtualización completa en la que el Hypervisor accede al sistema operativo directamente, a diferencia de la anterior que funcionaba como un intermediario (traductor). Es decir, la máquina virtual envía las instrucciones al procesador directamente, sin necesidad de ser traducidas. Así pues, la gestión del código máquina se realiza de una forma considerablemente más eficiente, al ejecutarse directamente, razón por la que el proceso de comunicación entre el hardware nativo y el sistema operativo de la MV es más eficiente que en el caso de la virtualización completa.

Dentro de esta técnica existen dos herramientas que la utilizan junto con la virtualización completa, como lo son: XEN y UML.

2.1.4.5.1 XEN.

Es la solución de fuente abierta creada en la Universidad de Cambridge que inicialmente ofrecía a los usuarios únicamente Paravirtualización a nivel del sistema operativo. Actualmente XEN es una solución de virtualización que nos ofrece virtualización completa únicamente bajo procesadores Intel VT o AMD Pacifica.

2.1.4.5.2 UML (User-mode LINUX).

Permite que un Linux ejecute otros sistemas operativos Linux en el espacio de usuario. Estos sistemas operativos alojados se ejecutan como un proceso alojado en el sistema Linux anfitrión. De esta forma sucede algo muy interesante, varios núcleos Linux se ejecutan en el contexto de un solo núcleo de Linux.

UML permite virtualizar dispositivos permitiendo así a los sistemas operativos alojados, compartir los dispositivos existentes: unidades CD-ROM, sistemas de ficheros, consolas, dispositivos NIC, etc. El núcleo anfitrión se ejecutará sobre el hardware y los núcleos alojados se ejecutan sobre el espacio de usuario del núcleo anfitrión. Los núcleos pueden ser anidados de forma que un núcleo alojado actúe como anfitrión de otro.

2.1.6 Beneficios de la Virtualización.

La tecnología de la virtualización nos trae un sin número de grandes ventajas como son:

- Reducción de los costes de espacio y consumo necesario.
- Administración global centralizada y simplificada.
- Rápida incorporación de nuevos recursos para los servidores virtualizados.
- Mejora en los procesos de clonación y copia de sistemas: Mayor facilidad para la creación de entornos de tests que permiten poner en marcha nuevas aplicaciones sin impactar a la producción, agilizando el proceso de las pruebas.
- No sólo aporta el beneficio directo en la reducción del hardware necesario, sino también los costes asociados.
- Aumento de la disponibilidad, y reducción de los tiempos de parada.
- Migración en caliente de máquinas virtuales (sin perder el servicio prestado) de un servidor físico a otro, eliminando la necesidad de paradas planificadas por mantenimiento de los servidores físicos.
- Alto grado de satisfacción general, y en especial por la reducción de los costes de administración.

- **Aislamiento**: las máquinas virtuales son totalmente independientes, entre sí y con el Hypervisor. Es decir, un fallo en una aplicación o en una máquina virtual afectará únicamente a esa máquina virtual. El resto de máquinas virtuales y el Hypervisor seguirán funcionando normalmente.
- **Seguridad**: cada máquina tiene un acceso de privilegio independiente. Por tanto, un ataque de seguridad en una máquina virtual sólo afectará a esa máquina.
- **Flexibilidad**: con la Virtualización, podemos crear máquinas virtuales con las características de CPU, memoria, disco y red que necesitemos, sin necesidad de "comprar" un ordenador con esas características. También podemos tener máquinas virtuales con distintos sistemas operativos, ejecutándose dentro de una misma máquina física.
- **Agilidad**: la creación de una máquina virtual es un proceso muy rápido y sencillo. Por tanto, si necesitamos un nuevo servidor lo podremos tener casi al instante, sin pasar por el proceso de compra, configuración, instalación, etc.
- **Portabilidad**: toda la configuración de una máquina virtual reside en uno o varios ficheros. Esto hace que sea muy fácil clonar o transportar la máquina virtual a otro servidor físico, simplemente copiando y moviendo dichos ficheros (que encapsulan o almacenan la máquina virtual) de un computador a otro.
- Mejora de las políticas de backup, recuperación ágil mediante puntos de control de las máquinas virtuales.
- Aprovechamiento óptimo de los recursos disponibles. Respuesta rápida ante cambios bajo demanda.
- Continuidad de negocio y recuperación ante desastres. En caso de fallo de un sistema físico, los sistemas lógicos allí contenidos pueden distribuirse dinámicamente a otros sistemas.
- Escalabilidad: Crecimiento ágil con contención de costes.
- Mantenimiento de aplicaciones heredadas. Aplicaciones propietarias que no han sido adaptadas a las nuevas versiones de sistema operativo.
- Virtual appliance - Aparato virtual: máquinas virtuales pre-configuradas, es decir, cargar y funcionar. Máquinas "ensambladas" y preconfiguradas para desempeñar una función determinada como por ejemplo: servidores de correo, bases de datos, aplicaciones cerradas, etc.

- Eficiencia energética. Implementar la tecnología de virtualización también juega un papel muy considerado en el ahorro de energía, y por ende para el bien del nuestro ecosistema.
- Alta disponibilidad, facilita de una manera más rápida la característica de Alta Disponibilidad que debe tener un Data Center de alto nivel y prestigio, es decir todos los servicios de TI dentro de una empresa funcionando a cada instante en cualquier momento a pesar de los contratiempos existentes. Los controles de calidad de cualquier empresa en estos momentos exigen Alta Disponibilidad en sus Departamentos de Sistemas.

En la Figura 4 se encuentra una lista de los beneficios que a menudo ayudan a las organizaciones IT a que justifiquen su migración hacia una infraestructura virtual. [*WILLIAMS, 2007]*

Categoria	Beneficio
Consolidación	Aumentar la utilización del servidor.
	Simplificar la migración de software legal.
	Anfitrión de sistemas operativos mezclados por la plataforma física.
	Optimice los entornos de prueba y desarrollo.
Confiabilidad	Aislar los fallos del software.
	Reasignar las particiones existentes.
	Crear particiones de conmutación por error dedicados o según sea necesario.
Seguridad	Contener los ataques digitales a través del aislamiento de fallos.
	Aplicar diferentes configuraciones de seguridad para cada partición.

Figura 4: Beneficios de la Virtualización.

Fuente: Virtualization with Xen, Pág. 14.

2.1.6.1 Consolidación.

El objetivo detrás de la consolidación es combinar y unificar. En el caso de la virtualización, las cargas de trabajo se combinan en menos plataformas físicas capaces de sostener la demanda de recursos, como CPU, memoria y E/S. En los centros de datos modernos, están lejos de utilizar al máximo el hardware en el cual funcionan, causando un desperdicio de infraestructura y una menor rentabilidad. Gracias a la consolidación, la virtualización permite combinar varias tareas del

servidor, o de los sistemas operativos y sus cargas de trabajo, de manera estratégica y colocarlos en el hardware compartido con la suficiente disponibilidad de recursos para satisfacer las demandas de estos. El resultado es una mayor utilización de los recursos disponibles. A menudo se piensa que los servidores no deben ser forzados a correr cerca de sus niveles de capacidad máxima. Sin embargo, es al contrario. Con el fin de maximizar la inversión, los servidores deben ejecutarse lo más cerca posible de su capacidad máxima, sin afectar a los trabajos en ejecución o los procesos de negocio. Con una buena planificación y la comprensión de los recursos a utilizar, la virtualización ayudará a aumentar el uso del servidor mientras que disminuye el número de plataformas físicas necesarias.

Otro beneficio de la consolidación se centra en la migración de sistemas heredados. Este ayuda a aliviar y simplificar las migraciones de los sistemas heredados, proporcionando una plataforma común y ampliamente compatible. Esto mejora las posibilidades de que las aplicaciones se puedan migrar a plataformas no compatibles y con riesgo a un hardware más nuevo, apoyados en un impacto mínimo.

2.1.6.2 Confiabilidad.

La confiabilidad tiene una relación directa con la disponibilidad del sistema, la aplicación del tiempo de actividad de los servicios y, en consecuencia, de la generación de ingresos. Las compañías deben estar dispuestas a realizar grandes inversiones en su infraestructura de servidores para asegurarse de que sus aplicaciones críticas del negocio permanezcan en línea y sus operaciones sean ininterrumpidas. Sus infraestructuras deben estar fortificadas para tolerar fallas e inactividad no planificada.

Las tecnologías de virtualización son sensibles en tratar estas áreas, proporcionando un gran aislamiento entre las máquinas virtuales en ejecución. Un fallo del sistema en una máquina virtual, o partición, no afectará a las otras particiones que se ejecuten en la misma plataforma de hardware. Este aislamiento protege y defiende las máquinas virtuales haciendo que estas no perciban lo ocurrido, y por lo tanto no se vean afectadas por las condiciones exteriores. Esta capa de abstracción es un componente clave en la virtualización, lo que hace que cada partición este ejecutándose como en un hardware dedicado.

2.1.6.3 Seguridad.

La misma tecnología que proporciona el aislamiento de fallos de las aplicaciones, también puede proporcionar el aislamiento de fallos de seguridad. En caso de estar en peligro una partición en particular, como se encuentra aislada del resto de las particiones, el fallo debe ser detenido antes de que se extienda a las demás. Otra solución puede ser implementar un aislamiento aún mayor para las particiones comprometidas e instancias del sistema operativo, al negarles los mismos recursos. Los ciclos de la CPU se pueden reducir, el acceso a la red y al disco de E/S cortó, o el sistema se haya detenido por completo. Estas tareas serían difíciles de llevar a cabo si la instancia comprometida se ejecuta directamente en un host físico.

2.1.7 Funcionamiento de la Virtualización.

El principal objetivo de la Virtualización es permitir que los sistemas operativos se ejecuten de forma independiente y de manera aislada, al igual que si se ejecuta directamente sobre hardware físico. Pero, ¿cómo se logra esto? Una gran cantidad de software de hipervisor y VMM han surgido para llevar a cabo la virtualización a través de mecanismos basados en software. Éstos crean un nivel de aislamiento de bajo nivel, donde el núcleo de la arquitectura del CPU trabaja más cerca de los niveles del software, para que cada máquina virtual tenga su propio entorno dedicado. De hecho, la relación entre la arquitectura de la CPU y los sistemas operativos virtualizados es la clave de cómo la virtualización en realidad trabaja con éxito.

2.1.7.1. Sistemas Operativos y el CPU.

El uso apropiado de las llamadas al sistema, requiere una cuidadosa coordinación entre el sistema operativo y el CPU. Ésta relación simbiótica entre sistema operativo y CPU proporciona muchas ventajas en la seguridad y la estabilidad. Un ejemplo de esto fue el sistema de time-sharing MULTICS, diseñado para una arquitectura de CPU especial.

MULTICS utilizó una división de las operaciones del software para eliminar el riesgo o la posibilidad de un colapso o inestabilidad de un componente que ha fallado y pueda afectar a otros

componentes. Se crearon los llamados anillos de protección, en lugar de separar el sistema operativo confiable, de los programas de usuario que no son de confianza. MULTICS incluyo ocho de estos anillos de protección, estos permitían diferentes niveles de abstracción y de aislamiento del núcleo central sin restricciones con el hardware.

La arquitectura de la CPU más común usada en las computadoras modernas es el IA-32, o compatible con arquitectura x86. Comenzando con el chipset 80286, la familia x86 proporciona dos métodos principales de direccionamiento de memoria: modo real y modo protegido. En el conjunto de chips 80386 y más tarde, un tercer modo se introdujo llamado modo virtual 8086 o VM86, que permitió la ejecución de programas escritos para el modo real, pero eludiendo las normas del modo real sin tener que subirlos a modo protegido. El modo real, se limita a un solo megabyte de memoria, estos se volvieron rápidamente obsoletas; y el modo virtual fue encerrado en al operación de 16 bits. El modo protegido fue la salvación para las x86, con numerosas y nuevas características para apoyar la multitarea. Éstas incluyen la segmentación de los procesos, por lo que ya no podían escribir fuera de su espacio de direcciones, junto con el soporte de hardware para la memoria virtual y multi-tarea. *[WILLIAMS, 2007]*

En la familia x86, el modo protegido utiliza cuatro niveles de privilegio, o anillos, numerados del 0 al 3. La memoria del sistema se divide en segmentos, y cada segmento se le asigna y dedica a un anillo en particular. El procesador utiliza el nivel de privilegios para determinar lo que se puede y no se puede hacer con el código o los datos dentro de un segmento. El Anillo-0 es el anillo más interno, con el control total del procesador. El Anillo-3 es el anillo exterior, con acceso más restringido, como se ilustran en la figura 5.

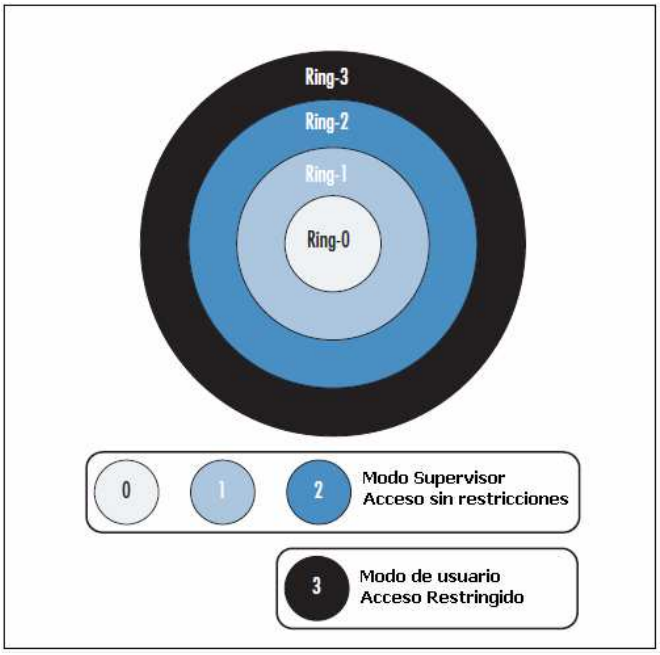

Figura 5: Anillos de privilegios de la arquitectura x86.

Fuente: Virtualization with Xen, Pág. 21.

NOTA:
El mismo concepto de anillos de protección es utilizado en la arquitectura de los sistemas operativos modernos. Windows, Linux, y la mayoría de las variantes de UNIX utilizan anillos, aunque han reducido la estructura de cuatro anillos a un enfoque de dos capas que utiliza sólo los anillos 0 y 3. El Anillo-0 es comúnmente llamado Modo Supervisor, mientras Anillo-3 es conocida como el modo de usuario. Los mecanismos de seguridad en el hardware hacen cumplir las restricciones sobre Anillo-3, al limitar el acceso a segmentos de código, paginación y entrada/salida. Si un programa de usuario que se ejecuta en el Anillo-3 trata de ingresar a la memoria fuera de sus segmentos, una alarma de proceso detiene la ejecución del código. Algunas instrucciones en lenguaje ensamblador no están aún disponibles para la ejecución fuera del Anillo-0, debido a su naturaleza de bajo nivel.

2.1.7.2 El Monitor de Máquina Virtual (VMM) y el Anillo-0.

Un procesador x86 se ejecuta en Modo Supervisor permitiendo la ejecución de todas las instrucciones, incluyendo las instrucciones privilegiadas, como las operaciones de E/S y de gestión de la memoria. Es en el modo Supervisor (Anillo-0) donde el sistema operativo normalmente ejecuta sus instrucciones. Cualquier inestabilidad del sistema en el Anillo-0 afecta directamente el

modo de usuario que se ejecuta en el Anillo-3. Con el fin de aislar el Anillo-0 para cada sistema virtualizado, se convierte necesario mover el Anillo-0 más cerca de estos clientes. De ésta manera, un fallo en el Anillo-0 para un huésped virtual no afecta al Anillo-0 del anfitrión, o en consecuencia al Anillo-3, de cualquier otro sistema virtual. El Anillo-0 no es perceptible para los demás huéspedes, estos podrán residir en los Anillos 1, 2, 3.

Dependiendo de su implementación, principalmente de cómo interactúan con la capa inmediatamente inferior y con la capa superior, los VMMs han sido clasificados en la historia reciente en dos tipos principales. Los VMM tipo 1 y los VMM tipo 2, la principal diferencia entre el tipo 1 y el tipo 2 radica en si el VMM es ejecutado directamente sobre el hardware nativo, de la misma forma que lo haría un kernel monolítico tradicional, o si es implementado como una aplicación que se ejecuta sobre un sistema operativo host. *[VON HAGEN. 2008]*

- VMM de Tipo 1 o Hipervisor: También denominado como unhosted. En este caso, el VMM corre directamente sobre el hardware nativo, teniendo que implementar sus propios drivers, módulo de manejo de memoria y planificación de la CPU. Es decir, se lo puede considerar como una aproximación de microkernel especializado para la ejecución de sistemas operativos invitados. El VMM puede ofrecer la totalidad o sólo algunas de las funcionalidades del hardware nativo a las máquinas virtuales que se ejecuten sobre el mismo. En la figura 6 se muestra al VMM como la primera capa de software ejecutándose sobre el hardware.

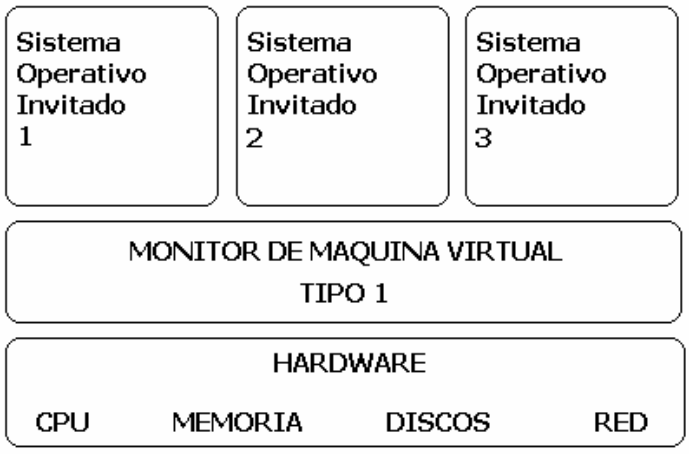

Figura 6: VMM de tipo 1.

Al tener acceso directo a los recursos de hardware, en lugar de tener que atravesar las capas de un sistema operativo, un hipervisor es más eficiente que su alternativa, un VMM tipo 2, y ofrece mayor escalabilidad, fiabilidad y mejor performance al reducir latencias e incrementar el rendimiento.

- VMM de tipo 2 o hosteado: En este caso el VMM se ejecuta sobre un sistema operativo que es el administrador real del hardware físico. En general el VMM de tipo 2 se implementa como un proceso tradicional del sistema operativo host. Por lo tanto, debe capturar las invocaciones a funcionalidades del hardware que realicen las máquinas virtuales y traducirlas a invocaciones reales al sistema operativo host, el cuál a su vez debe transformar en llamadas a funciones del hardware nativo.

  Esta clara separación entre máquina virtual y hardware subyacente tiene la ventaja de facilitar la implementación y permitir un amplio abanico de posibilidades a la hora de ofrecer instancias virtuales de recursos o dispositivos de hardware virtualizados.
  Sin embargo, los altos costos de performance de éstas soluciones, comparadas con la alternativa del VMM tipo 1, hacen que en los entornos de producción no se utilice esta forma de implementación. Debe tenerse en cuenta que cada intento de acceso al hardware subyacente por parte de una máquina virtual deberá atravesar más dominios de protección y generará más cambios de contexto que en el caso anterior.

  En comparación con los VMM de tipo 1, los de tipo 2 se consideran menos confiables, menos escalables, mucho menos eficientes y con latencias considerables. La gran ventaja radica en que pueden ser ejecutados como un simple programa, razón por la cuál un usuario final suele preferir instalar esta variante en su estación de trabajo cuando sus objetivos no incluyen requerimientos de escalabilidad o alta eficiencia. La figura 7 muestra un VMM de tipo 2.

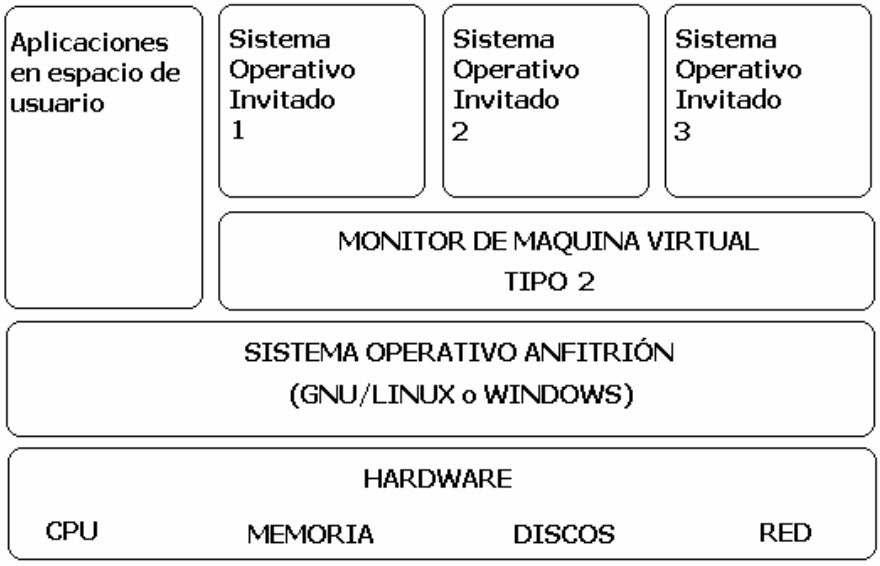

Figura 7: VMM de tipo 2.

2.1.7.3 Los Requisitos Popek y Goldberg:

Los Requerimientos de virtualización de Popek y Goldberg son un conjunto de condiciones suficientes para que una arquitectura de computadoras soporte eficientemente la virtualización. Fueron elaborados por Gerald J. Popek y Robert P. Goldberg en su artículo de 1974 “Formal Requirements for Virtualizable Third Generation Architectures”. *[WILLIAMS, 2007]*

Aunque los requisitos se derivan de suposiciones simplificadas, todavía constituyen una manera eficaz de determinar si una arquitectura soporta eficientemente la virtualización, y proporcionan líneas maestras para el diseño de arquitecturas virtualizables. El artículo presenta tres propiedades de interés cuando se analiza el entorno creado por un VMM:

- **Equivalencia / Fidelidad**: un programa ejecutándose sobre un VMM debería que tener un comportamiento idéntico al que tendría ejecutándose directamente sobre el hardware subyacente.
- **Control de recursos / Seguridad**: El VMM tiene que controlar completamente y en todo momento el conjunto de recursos virtualizados que proporciona a cada invitado.

- **Eficiencia / Performance**: Una fracción estadísticamente dominante de instrucciones tienen que ser ejecutadas sin la intervención del VMM, o en otras palabras, directamente por el hardware.

Según la terminología de Popek y Gordberg, un VMM debe presentar cada una de las tres propiedades. Se asume típicamente que los VMM satisfacen la fidelidad y seguridad, y aquellos VMMs que adicionalmente satisfacen la propiedad de performance son llamados VMMs eficientes.

Popek y Goldberg describen las características que el Conjunto de Instrucciones de la Arquitectura (ISA, por las siglas de Instruction Set Architecture) de la máquina física debe poseer para poder ejecutar VMMs que tengan las propiedades listadas arriba. Su análisis deriva tales características usando un modelo de "arquitectura de tercera generación" (por ejemplo, IBM 360, Honeywell 6000, DEC PDP-10) que, de cualquier forma, es lo suficientemente general como para ser extendido a las máquinas modernas. Este modelo incluye un procesador que opera tanto en modo usuario como en modo privilegiado y tiene acceso a una memoria lineal uniformemente direccionable. Se asume que un subconjunto del ISA está disponible sólo cuando el procesador se encuentra en modo privilegiado y que la memoria es direccionada en términos relativos a un registro de relocación. Las interrupciones y la Entrada/Salida no están modeladas.

2.1.7.3.1 Teoremas de Virtualización.

Para deducir sus teoremas de virtualización, que dan condiciones suficientes (pero no necesarias) para la virtualización, Popek y Goldberg introducen una clasificación de las instrucciones de un ISA en tres grupos diferentes:

- **Instrucciones privilegiadas**: son instrucciones que provocan una excepción al ser ejecutadas en modo usuario y no la provocan al ser ejecutadas desde modo supervisor.
- **Instrucciones sensibles de control**: éstas son instrucciones que permiten cambiar la configuración actual de los recursos hardware.
- **Instrucciones sensibles de comportamiento**: son instrucciones cuyo comportamiento o resultado dependen de la configuración del sistema. Por esto mismo, permiten conocer el estado de la configuración actual de los recursos hardware.

El resultado principal de Popek y Goldberg puede ser expresado de la siguiente forma:

Teorema 1: Para cualquier computadora convencional de la tercera generación, puede construirse un VMM si el conjunto de instrucciones sensibles (tanto de control como de comportamiento) para esa computadora es un subconjunto del de las instrucciones privilegiadas.

Este teorema establece que para construir un VMM es suficiente con que todas las instrucciones que podrían afectar al correcto funcionamiento del VMM (instrucciones sensibles) siempre generen una excepción y pasen el control al VMM. Eso garantiza la propiedad de control de los recursos y la seguridad. En cambio, las instrucciones no privilegiadas deben ejecutarse nativamente (es decir, eficientemente).

Este teorema también provee una técnica simple para implementar un VMM, llamada virtualización por “trap and emulate” (excepción y emulación), más recientemente llamada virtualización clásica: ya que todas las instrucciones sensibles se comportan correctamente, todo lo que un VMM tiene que hacer es ‘atrapar’ y emular cada una de ellas.

Un problema relacionado es el de derivar condiciones suficientes para la virtualización recursiva, es decir, las condiciones bajo las cuales puede construirse un VMM capaz de ejecutar una copia de si mismo.

Teorema 2: Una computadora convencional de la tercera generación es recursivamente virtualizable si:

1. es virtualizable y,
2. puede construirse un VMM sin dependencias de tiempo para ella.

2.1.7.3.2 Técnicas de la virtualización clásica.

En esta sección se describen las ideas más importantes usadas en las implementaciones de los VMM clásicos: deprivileging, estructuras shadow y trazas.

2.1.7.3.2.1 Deprivileging

En una arquitectura clásicamente virtualizable, todas las instrucciones que se leen o escriben en estado privilegiado pueden generar una excepción cuando se ejecutan en un contexto no privilegiado. A veces, las excepciones ocurren por la clase de la instrucción (por ejemplo, una instrucción OUT), y otras veces las excepciones suceden como resultado de la protección por parte del VMM de ciertas estructuras en memoria que la instrucción intenta acceder (por ejemplo, el rango de direcciones de un dispositivo de E/S mapeado en memoria).

Un VMM clásico ejecuta directamente sobre el hardware a los sistemas operativos invitados, pero en un nivel de privilegios reducido. El VMM intercepta las excepciones del invitado con menos privilegios, y emula, sobre el estado de la máquina virtual, la instrucción generadora de la excepción.

2.1.7.3.2.2 Estructuras primarias y estructuras shadow.

El estado privilegiado de un sistema virtual difiere del estado del hardware subyacente. La función básica del VMM es proveer un entorno de ejecución que cumpla con las expectativas del invitado, a pesar de aquella diferencia. Para lograr esto, el VMM deriva estructuras secundarias (más conocidas como estructuras shadow) de las estructuras primarias del invitado.

2.1.7.3.2.3 Trazas de memoria.

Para mantener la coherencia de las estructuras shadow, los VMMs usan mecanismos por hardware de protección de páginas para interceptar accesos a las estructuras primarias en memoria. Por ejemplo, las entradas a las tablas de páginas del invitado, para las cuales se han construidos entradas de páginas shadows pueden ser protegidas contra escritura. Los dispositivos mapeados en memoria generalmente deben ser protegidos tanto contra lectura como contra escritura. Esta técnica basada en protección de página es conocida como traceado (tracing). Los VMMs clásicos manejan un fallo de traza de manera similar a un fallo por instrucción privilegiada: decodifican la instrucción del invitado que generó la excepción, emulan su efecto sobre la estructura primaria y propagan el cambio a la estructura shadow.

2.2 Paravirtualización.

La paravirtualización, al igual que las técnicas de virtualización alternativas utilizadas en la actualidad, es una idea que data de los primeros intentos de sistemas operativos de tiempo compartido para mainframes, como se mencionó en el capítulo 2.1.4.5.

La paravirtualización permite que múltiples sistemas operativos se ejecuten en el hardware al mismo tiempo haciendo un uso eficiente de los recursos, como el procesador y la memoria, a través de la multiplexación efectiva de esos recursos entre las máquinas virtuales. *[TAKENURA, 2010]*

A diferencia de la virtualización completa, donde se emula todo el sistema (BIOS, discos, procesadores, NICs, etc), un VMM de paravirtualización opera con invitados que han sido modificados para cooperar. En la figura 8 observamos el funcionamiento de la paravirtualización.

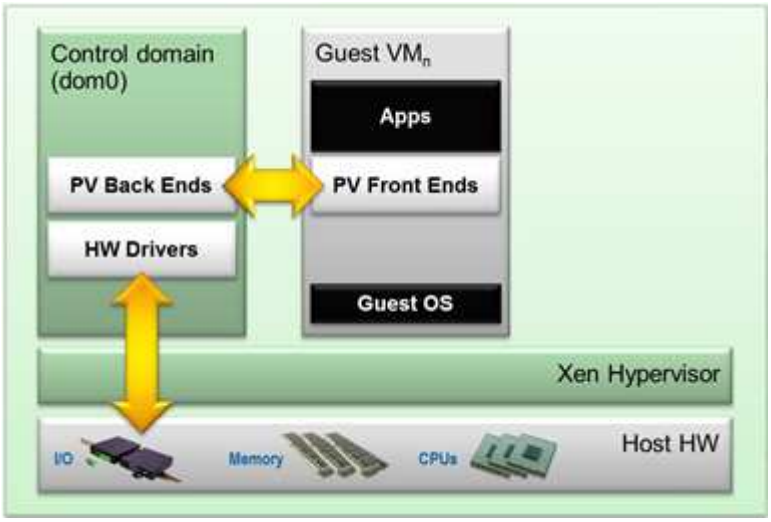

Figura 8: Funcionamiento de la Paravirtualización.
Fuente: wiki.xenproject.org.

Como se explico anteriormente, el kernel de un sistema operativo monolítico se suele ejecutar en el nivel 0 (más privilegiado) mientras que las aplicaciones corren en el nivel 3 (menos privilegiado). En la paravirtualización también se utiliza la técnica de anillos-deprivileging, por la cual el sistema operativo es modificado para poder ejecutarse en el nivel 1, dejando el nivel 0 para una ejecución segura del hipervisor.

La característica distintiva radica en la nueva forma de comunicación entre el sistema operativo invitado y el VMM para mejorar la performance. El código fuente del kernel invitado debe modificarse para reemplazar las instrucciones no virtualizables por llamadas explícitas al VMM, denominadas hypercalls. Se elimina de esta forma la necesidad de utilizar excepciones, que suelen ser costosas en términos de performance, o de utilizar traducción binaria, que son complejas de implementar y mantener.

La interfaz de la máquina virtual difiere de la interfaz ofrecida por el hardware subyacente. Es decir las hypercalls son distintas de las instrucciones nativas, motivo por el cuál se requiere la modificación del código fuente del kernel invitado. El hipervisor también puede proveer interfaces adicionales de hypercalls para otras operaciones críticas del kernel como el manejo de memoria, el manejo de interrupciones y el mantenimiento del tiempo.

Es por esta comunicación explícita que se suele decir, que el sistema operativo invitado “sabe” que está siendo virtualizado y que, colabora con el hipervisor con mecanismos de comunicación de más alto nivel que en el caso de la traducción binaria. Esta abstracción generalmente implica una mejor performance que en los modelos de virtualización completa, donde cada componente debe ser emulado y la comunicación es de extremado bajo nivel.

Incluso aunque es muy difícil construir las sofisticadas técnicas de traducción binaria adaptativas necesarias para una eficiente virtualización completa, resulta relativamente sencillo, modificar el código fuente del sistema operativo invitado para la implementación de la paravirtualización, reduciendo la complejidad de la capa de virtualización.

Además de la mayor performance alcanzada, la eficiencia de esta técnica puede traducirse en una mayor capacidad para escalar. Asumiendo un costo realista para cada técnica de virtualización, si una solución de virtualización completa requiere un 10% de utilización de la CPU por instancia del invitado, por procesador para implementar la máquina virtual, y a la vez cada invitado necesita utilizar un 10% el tiempo de CPU para cálculos útiles, entonces como vemos en el cuadro de la figura 9, que sólo podremos ejecutar cinco instancias de invitado, dada la plataforma de hardware.

	Instancias de Invitados	Costo de la virtualización	Procesamiento útil	Total
Virtualización Completa	5	10% (50% total)	10% (50% total)	100%
Paravirtualización	8	2% (16% total)	10% (80% total)	96%

Figura 9: Comparación entre Virtualización Completa y Paravirtualización.
Fuente: wiki.xenproject.org, Abril 2013.

El costo adicional de la paravirtualización suele estar entre el 0.1% y el 3% de tiempo de CPU por invitado. Tomamos en consideración una media aproximada de 2%. Con lo cual, en este caso, podríamos ejecutar ocho instancias de invitados que requirieran un 10% de tiempo de cómputo útil.

Sin embargo, esta eficiencia tiene un costo asociado en cuanto a flexibilidad y seguridad. La flexibilidad se reduce notablemente porque es necesario modificar un sistema operativo para que pueda ejecutarse como invitado. La modificación que deben sufrir los sistemas operativos presenta una limitación ya que sólo puede llevarse a cabo en sistemas operativos de código abierto (o sólo por el fabricante del software, en el caso del software propietario). Por lo tanto, en un comienzo sólo GNU/Linux y algunas variantes de BSD fueron portados para poder ejecutarse en hipervisores.

Más recientemente, sin embargo, esta técnica fue adoptada por productos de virtualización comerciales y sistemas operativos de código cerrado para resolver ciertas cuestiones específicas, como la virtualización de dispositivos.

Resumiendo lo anterior, la paravirtualización es el acto de ejecutar un sistema operativo invitado, bajo el control de un sistema anfitrión, donde el invitado ha sido portado a una arquitectura virtual que es muy parecida, pero no igual, a la arquitectura del hardware sobre el cual está corriendo realmente. Esta técnica permite implementar virtualización de manera eficiente.

2.2.1 Xen.

2.2.1.2 ¿Qué es Xen?

El proyecto Xen se originó como un proyecto de investigación del Grupo de Investigación de Sistemas de la Universidad de Cambridge Computer Laboratory. Fue bautizado como el

Proyecto XenoServers, y fue financiado por Consejo de Investigación de Ingeniería y Ciencias Físicas del Reino Unido (EPSRC). El objetivo del proyecto fue proporcionar una infraestructura pública a nivel mundial y que sea accesible para los propósitos de área amplia de computación distribuida. Con una especial dedicación a la investigación de sistemas, y dirigido por el investigador Ian Pratt, el proyecto creo el hipervisor Xen como su tecnología principal. [VON HAGEN, 2008].

Xen es un hipervisor tipo 1 o baremetal de código abierto, lo cual hace posible ejecutar muchas instancias de un sistema operativo o incluso diferentes sistemas operativos en paralelo en una sola máquina (o anfitrión). Xen es el único hipervisor tipo 1 que está disponible como código abierto. Se utiliza como base para una serie de aplicaciones comerciales y de código abierto, tales como: la virtualización de servidores, Infraestructura como Servicio (IaaS), la virtualización de escritorios, aplicaciones de seguridad y dispositivos de hardware.

Estas son algunas de las ventajas claves de Xen:

- **Ocupa poco espacio y la interfaz es liviana** (de alrededor de 1 MB de tamaño). Debido a Xen utiliza un diseño de microkernel, con una pequeña cantidad de memoria y una interfaz limitada para el cliente, es más robusto y seguro que otros hipervisores.
- **Independiente del sistema operativo**: La mayoría de las instalaciones funcionan con Linux como la pila de control principal (también conocido como "dominio 0"). Sin embargo, otros sistemas operativos pueden utilizar XEN, incluyendo NetBSD y OpenSolaris.
- **Aislamiento de drivers**: Xen tiene la capacidad de permitir que el driver del dispositivo principal de un sistema pueda ejecutarse dentro de una máquina virtual. Si se bloquea el driver, o se ve comprometida, la máquina virtual que contiene el driver puede ser reiniciado y el driver reiniciado no afecta al resto del sistema.
- **Paravirtualización**: huéspedes totalmente virtualizados se han optimizado para funcionar como una máquina virtual. Esto permite a los clientes que se ejecuten mucho más rápido que con las extensiones de hardware. Además, Xen puede ejecutarse en hardware que no admite extensiones de virtualización.

2.2.1.3 Componentes de Xen.

A continuación se detallan los componentes principales de Xen, los cuales se muestran en la figura 10.

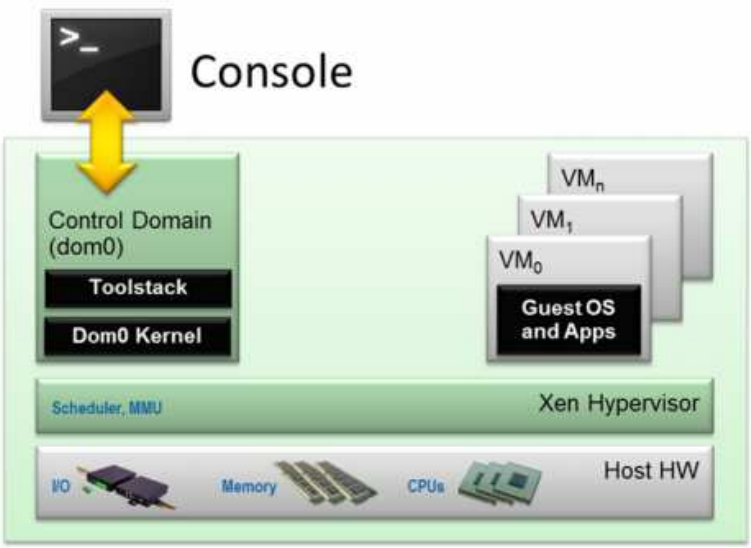

Figura 10: Componentes de Xen. Fuente: wiki.xenproject.org.

El hipervisor Xen: es una capa de software excepcionalmente delgada (menos de 150.000 líneas de código) que se ejecuta directamente en el hardware y se encarga de la gestión de la CPU, la memoria y las interrupciones. Es el primer programa que se ejecuta después de salir del gestor de arranque. El hipervisor en sí no tiene conocimiento de las funciones de E/S, tales como la creación de redes y almacenamiento.

Dominios invitados/Máquinas virtuales son entornos virtuales, cada uno ejecuta su propio sistema operativo y las aplicaciones. Xen es compatible con dos modos diferentes de virtualización:

- Paravirtualización (PV) y
- Virtualización asistida por hardware o Full (HVM).

Ambos tipos de virtualización se pueden utilizar al mismo tiempo en un solo sistema Xen. También es posible utilizar técnicas utilizadas para la Paravirtualización en un huésped HVM, esencialmente creando una continuidad entre la PV y HVM. Este enfoque se denomina PV en HVM. Los huéspedes Xen son totalmente aislados del hardware, en otras palabras, no tienen el privilegio de acceder al hardware o las funcionalidades de E/S. Por lo tanto, también se les llama dominio sin privilegios (o DomU).

El dominio de control (o dominio 0) es una máquina virtual con privilegios especiales, como la capacidad de acceder al hardware directamente, se encarga de todos los accesos a las funciones del sistema de E/S e interactúa con las otras máquinas virtuales. También expone una interfaz de control con el mundo exterior, a través del cual se controla el sistema. El hipervisor Xen no funciona sin dominio 0, que es la primera VM iniciada por el sistema.

Toolstack y consola del dominio 0, contiene una pila de control (llamada Toolstack) que permite al usuario gestionar la creación de la máquina virtual, la destrucción y la configuración. El toolstack expone una interfaz que, o bien es impulsado por una consola de línea de comandos, por una interfaz gráfica o por una pila tal como OpenStack o CloudStack.

Sistemas operativos habilitados para Xen: Un dominio 0 necesita un kernel Xen. Los huéspedes virtualizados requieren un kernel PV-enabled. Las distribuciones de Linux basadas en los kernels de Linux más recientes son Xen-habilitados y por lo general contienen los paquetes que contienen el Xen Hypervisor y herramientas Xen. Con distribuciones Windows es necesaria el Toolstack por defecto y una configuración por consola.

2.2.1.4 Características de Xen.

Xen ofrece un potente conjunto de funcionalidades que son adecuadas para las implementaciones de aplicaciones necesarias tanto para las pequeñas y medianas empresas. Éstas características incluyen:

- Máquinas virtuales con un rendimiento casi nativo.
- Soporte completo en x86 (32 bits), x86 (32 bits) con Extensión de Dirección Física (PAE), y x86 con extensiones de 64 bits.
- Soporte para casi todo el hardware de controladores Linux disponibles.
- Múltiples CPU virtuales soportados en cada equipo invitado.
- La asignación dinámica de recursos a través de CPU’s virtuales, de conexión en caliente (si el sistema operativo invitado lo soporta).
- La migración sin tiempo de inactividad de máquinas virtuales en ejecución entre dos hosts físicos.

- Soporte de hardware de ayuda de los procesadores de Intel (Intel-VT) y AMD (AMD-V), permitiendo que los sistemas operativos invitados no sean modificados.

2.2.1.5 Arquitectura del Procesador.

Xen es compatible actualmente con una amplia dotación de hardware y sistemas operativos. Parte de su propuesta es ser una mercancía de virtualización de los productos más básicos, lo que le permite virtualizar sistemas operativos populares. Aunque Xen fue desarrollado originalmente como un proyecto de VMM para x86, los nuevos sistemas están siendo explorados y probados para plataformas adicionales, incluyendo Itanium de Intel e Itanium 2 (IA-64) y PowerPC de IBM.

2.2.1.6 Paravirtualización con Xen.

Xen presenta una abstracción de máquina virtual que es muy similar a la plataforma de hardware, sin crear una copia exacta de la misma. Esta técnica es el corazón de lo que se llama paravirtualización. *[WILLIAMS, 2007]*

Para lograr esto, la paravirtualización requiere cierta modificación en el sistema operativo invitado para que sea consciente con la capa de virtualización subyacente, mejorar su interacción tanto con el mundo virtual que ha sido presentada, así como el mundo real que se ejecuta en la parte superior. Aunque las aplicaciones binarias no deben ser modificas también, las modificaciones del sistema operativo facilitan un mejor rendimiento y realizan un mejor manejo de las operaciones. La Paravirtualización tiene ciertos requisitos para la gestión de memoria, CPU, y dispositivos de E/S, estos requisitos se detallan en la figura 11.

Tipo de Elemento	Elemento	Requisitos o Consideración especial
Gestión de memoria	Segmentación	No se puede insertar descriptores de segmentos privilegiados y no puede solaparse con el extremo superior del espacio de direcciones lineal.
	Paginación	El sistema operativo huésped tiene acceso directo de lectura a las tablas de páginas de nivel de hardware, pero las actualizaciones se procesan por lotes o realizar individualmente y validado por el hipervisor.
CPU	Protección	El sistema operativo invitado debe funcionar a un nivel de privilegio más restringido de Xen en otras palabras, no se puede ejecutar en el Ring-0.
	Excepciones	El sistema operativo invitado debe registrar una tabla para atrapar los manipuladores de excepciones.
	Las llamadas al sistema	El sistema operativo invitado puede instalar un controlador para las llamadas al sistema, permitiendo que las llamadas directas de una aplicación o del sistema operativo en sí. Algunas de estas llamadas no tienen que ser manejadas directamente por Xen.
	Interrupciones	Las interrupciones de hardware se sustituyen por un mecanismo de notificación de eventos.
	Tiempo	El sistema operativo invitado debe ser consciente y tener en cuenta a la vez el tiempo real, el tiempo de reloj, así como el tiempo virtual.
Dispositivos de E/S	Red y disco	Los dispositivos virtuales son fáciles de acceder. Los datos se transfieren mediante E/S asíncrona de anillos, e interrumpir la comunicación con el sistema operativo invitado que se maneja a través de las notificaciones de eventos.

Figura 11: Requisitos y Consideraciones de la Paravirtualización.

Fuente: Virtualization with Xen, Pág. 53.

2.2.1.8 Dominios Xen.

En la figura 14 observamos la estructura de un sistema típico de virtualización completa para una arquitectura x86.

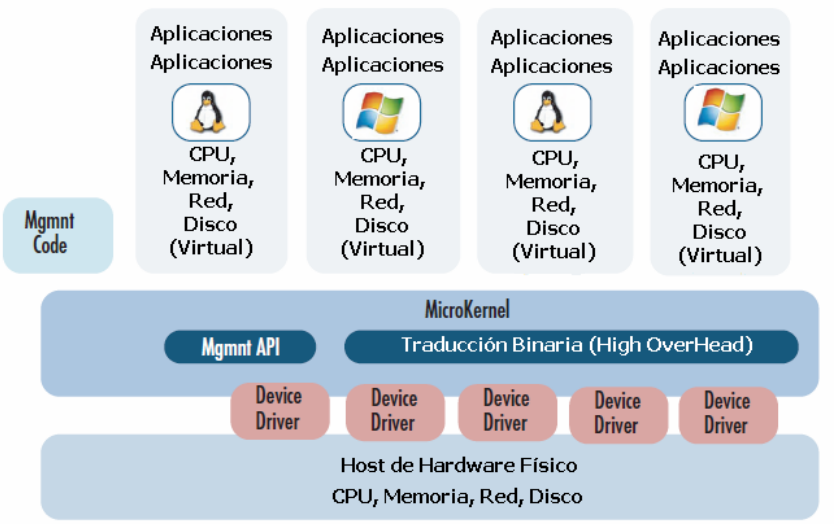

Figura 14: Estructura típica de sistema de virtualización completa para x86.

Fuente: Virtualization with Xen, Pág. 54.

Aunque similar en función a otro MMV, Xen tiene una estructura única de sistema que se descompone en: el hardware subyacente, el Hipervisor, una región de control, y las propias máquinas virtuales, como podemos ver en la figura 15.

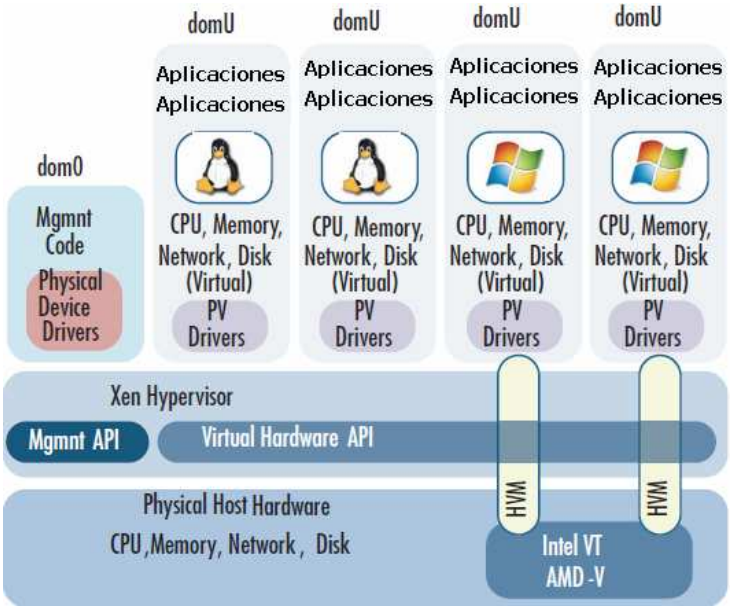

Figura 15: Estructura del sistema de la MMV Xen.

Fuente: Virtualization with Xen, Pág 55.

El hipervisor expone una capa de abstracción que contiene tanto la Administración como las API del hardware virtual, incluyendo una interfaz de control que permite a los clientes interactuar directa e indirectamente con el hardware subyacente. El Administrador de API's es una región de control que tiene acceso privilegiado al hardware y contiene código de gestión de modo de usuario. Tanto la región de control y las máquinas virtuales se conocen como "dominios". Al arrancar, el sistema inicia un dominio especial donde está permitido el uso de la interfaz de control, denominado Dominio 0 o dom0. Este dominio inicial aloja el software de gestión del modo de usuario que se utiliza para administrar el entorno de Xen, ya sea a través de línea de comandos o usando la consola de administrador de XenSource. También es responsable de iniciar y detener un tipo de dominio con menos privilegios, los dominios invitados denominados Dominio-U o domU, a través de la interfaz de control, así como el control de la CPU, las asignaciones de memoria, y el acceso a los dispositivos tales como almacenamiento en disco y las interfaces de red física. A los domU también se los conoce como una máquina virtual Xen, o XenVM. *[WILLIAMS, 2007]*

El dominio de control, dom0, funciona como una instalación estándar de Linux. Puede ejecutar aplicaciones en modo de usuario, tales como las que se utilizan para gestionar el entorno Xen, así como instalar los controladores de dispositivos necesarios para soportar su plataforma de hardware. Debido a la capacidad para compilar y ejecutar prácticamente cualquier dispositivo de hardware con los controladores de Linux disponibles, Xen tiene una amplia gama de hardware soportado. Esto permite a las organizaciones de IT una mayor flexibilidad en su selección de dispositivos de red y de almacenamiento físico y permiten a Xen ser implementado en casi cualquier entorno x86.

2.3 Windows Server 2003.

Windows Server 2003 es un sistema operativo de propósitos múltiples capaz de manejar una gran gama de funciones de servidor, en base a las necesidades, tanto de manera centralizada como distribuida. Algunas de estas funciones del servidor son:

- Servidor de archivos e impresión.
- Servidor Web y aplicaciones Web.
- Servidor de correo.

- Terminal Server.
- Servidor de acceso remoto/red privada virtual (VPN).
- Servicio de directorio, Sistema de dominio (DNS), y servidor DHCP.
- Servidor de transmisión de multimedia en tiempo real (Streaming).
- Servidor de infraestructura para aplicaciones de negocios en línea (tales como planificación de recursos de una empresa y software de administración de relaciones con el cliente).

2.3.1 Características.

Sus características más importantes son:

- Sistema de archivos NTFS:
  1. Cuotas
  2. Cifrado y compresión de archivos, carpetas y no unidades completas.
  3. Permite montar dispositivos de almacenamiento sobre sistemas de archivos de otros dispositivos al estilo unix
- Gestión de almacenamiento, backups... incluye gestión jerárquica del almacenamiento, consiste en utilizar un algoritmo de caché para pasar los datos menos usados de discos duros a medios ópticos o similares más lentos, y volverlos a leer a disco duro cuando se necesitan.
- Windows Driver Model: Implementación básica de los dispositivos más utilizados, de esa manera los fabricantes de dispositivos sólo han de programar ciertas especificaciones de su hardware.
- ActiveDirectory; Directorio de organización basado en LDAP, permite gestionar de forma centralizada la seguridad de una red corporativa a nivel local.
- Autentificación Kerberos5.
- DNS con registro de IP's dinámicamente.
- Políticas de seguridad.

2.4 TrixBox.

Es una distribución del sistema operativo GNU/Linux, basada en CentOS, que tiene la particularidad de ser una central telefónica (PBX) por software basada en la PBX de código abierto Asterisk. Como cualquier central PBX, permite interconectar teléfonos internos de una compañía y conectarlos a la red telefónica convencional (RTB - Red telefónica básica). La versión Trixbox CE es la continuación de Asterisk At Home.

El paquete trixbox incluye muchas características que antes sólo estaban disponibles en caros sistemas propietarios como creación de extensiones, envío de mensajes de voz a e-mail, llamadas en conferencia, menús de voz interactivos y distribución automática de llamadas.

Al ser un software de código abierto, posee varios beneficios, como es la creación de nuevas funcionalidades. Algo muy importante es que no sólo soporta conexión a la telefonía tradicional, sino que también ofrece servicios VoIP -voz sobre IP-, permitiendo así ahorros muy significativos en el coste de las llamadas internacionales, dado que éstas no son realizadas por la línea telefónica tradicional, sino que utilizan Internet. Los protocolos con los cuales trabaja pueden ser SIP, H.323, IAX, IAX2 y MGCP.

Trixbox se ejecuta sobre el sistema operativo CentOS y está diseñado para empresas de 2 a 50 empleados.

2.4.1 TrixBox CE (Community Edition).

Comenzó en el año 2004 como un proyecto popular IP-PBX denominado Asterisk@Home. Desde ese momento se convirtió en la distribución más popular, con más de 65.000 descargas al mes. Dicha versión se caracteriza por dos pilares importantes: su flexibilidad para satisfacer las necesidades de los clientes y, sobre todo, por ser gratuita.

2.4.2 ¿Por qué utilizar TrixBox CE?

Tal como se dijo anteriormente TrixBox CE es una versión muy flexible, que no solo permite configurar funciones y módulos parametrizables para las necesidades de cada cliente, sino que también es posible acudir a la comunidad de TrixBox para ayudar o ser ayudado. Ésta es una de

las más grandes y más activas del mundo y sus miembros trabajan entre ellos día a día con el fin de responder consultas, resolver problemas, fallos y en seguir desarrollando la herramienta.

2.5 Hardware requerido para virtualizar.

Sin meterse a un despliegue profesional y a modo de una recomendación propia, con un CPU de dos o cuatro núcleos, 1 a 4 GB de RAM, un disco duro de 300 a 500 GB y tarjeta de red Gigabit Ethernet es suficiente para montar un servidor de máquinas virtuales pequeño. Que quiero decir con esto, los servidores que se instalaron en este proyecto no necesitan ni mucho espacio de disco o memoria para funcionar correctamente. Pero para una empresa que requiera de servicios de mayor consumo y utilidad serían necesarios servidores con capacidades superiores.

Al buscarse el hardware adecuado para la virtualización, hay algunas cosas a tener en cuenta:

- **Memoria RAM**: en este proyecto se asigna a cada servidor virtualizado 1 GB de memoria RAM. Pero dependiendo del funcionamiento y del servidor que se virtualize, es la cantidad de memoria que se deberá asignar a cada uno de ellos. Este es un elemento clave, mientras que contemos con más memoria mejor.
- **Almacenamiento**: para servidores que necesitan mucho espacio de almacenamiento como ser las diversas versiones de Windows Server, donde se utilizan servicios de carpetas compartidas, o escritorios remotos, éstos pueden llegar a utilizar desde Gigabits hasta Terabits en grandes empresas. Mientras que un servidor de telefonía IP, no requiere de gran tamaño de disco ya que solo administra usuarios y sus llamadas. Por ello hay que tener mucho cuidado a la hora de la asignación de espacios de disco duro.
- **Procesador**: como se mencionó anteriormente, contar con más de dos núcleos, sería necesario para el host anfitrión. Pero el problema se plantea con la paravirtualización, sería necesario además contar con un procesador INTEL-VT o un AMD-V para poder soportar esta tecnología de virtualización.
- **Red**: para no generar una demora en esta parte del sistema sería bueno contar con dos puertos de red que preferentemente sean Gigabit Ethernet.

Estos puntos son los más importantes a considerar a la hora de elegir el hardware para la implementación de un servidor en el que operen varias maquinas virtuales.

2.6 Software para la virtualización de sistemas.

Una vez que disponemos del equipo adecuado, la segunda parte consiste en elegir la plataforma de virtualización que más se adapte a las necesidades de la empresa. Las opciones a elegir son muchas, a continuación se nombran una gran mayoría de los sistemas de virtualización actuales:

Software de virtualización gratuito (Open Source):

- QEmu para Windows, Solaris, Linux, FreeBSD, NetBSD, OpenBSD, Mac OS X, ZETA, BeOS.
- VirtualBox Windows, Linux, Mac OS X.
- XEN para Linux, Unix-like, BSD, OpenSolaris.
- OpenVZ para Linux.

Software de virtualización gratuito (Freeware):

- VMWare Server para Windows y Linux.
- VMWare Player para Windows.
- Microsoft Virtual Server 2005 para Windows.

Software de virtualización de pago:

- Parallels Desktop para Mac OS X.
- Parallels Workstation para Windows y Linux.
- VMWare Fusion para Mac OS X.
- VMware Workstation para Windows y Linux.

Sistemas Operativos para virtualización de pago:

A diferencia con los anteriores, éstos se instalan directamente sobre el hardware sin necesidad de un sistema operativo anfitrión.

- VMware ESX Server 3.
- Virtuozzo basado en el mencionado OpenVZ.
- Virtual Iron VFe.

Capítulo 3. Desarrollo

3.1 La Idea.

La idea de este proyecto surge de observar el crecimiento de las Pymes y su necesidad de contar con diversos servicios y aplicaciones informáticas disponibles tanto para sus clientes como sus propios empleados. Nace de una necesidad real y se muestra como una solución a un problema que no sólo afecta a las Pymes sino también a grandes empresas que aún no optan por una solución a estas necesidades.

La situación actual que viven los responsables de tecnología dado que dedican un servidor físico a cada tipo de aplicación (servidores Web, servidor de aplicaciones, Exchange, bases de datos), produciendo un “Server Sprawl” (despliegue desbordado de servidores o proliferación de servidores) y sus demás desventajas que esto ocasiona.

Actualmente podemos evitar este problema con las nuevas tecnologías de hardware que hacen inadecuada esta metodología de servidores dedicados. A su vez se tiende cada día a la consolidación y optimización de recursos aunque aun se tiene en cuenta obviamente el costo de hardware debido a que es un factor relevante.

Para enfocar la solución a esta problemática se utiliza una técnica de virtualización a nivel de sistema operativo, para aprovechar al máximo los recursos de hardware brindados, utilizando poderosas herramientas y un sistema operativo como base y punto de partida de este trabajo de tesis.

La implementación de estas dos técnicas de virtualización y la realización de pruebas en ambas, permitirá presentar una comparación fehaciente y descriptiva de que método utilizar a la hora de necesitar la empresa de incursionar en este camino.

3.2 Hardware a utilizar.

El equipo que fue utilizado como servidor host para sendas técnicas de virtualización se detalla a continuación:

- Tipo de CPU: DualCore AMD Athlon 64 X2, 2600 MHz (13 x 200) 5200+
- Nombre del motherboard: Asus M2N32-SLI Deluxe
- Memoria del sistema: 3072 MB (DDR2-800 DDR2 SDRAM)
- Disco rígido: SAMSUNG HD502HJ (500 GB, 7200 RPM, SATA-II)
- Placa de video: NVIDIA GeForce 8500 GT (512 MB)
- Tarjeta de Red: nVIDIA nForce 590 SLI (MCP55PXE) - Gigabit LAN Controller

3.3 Software a utilizar.

En el servidor host que prestara los servicios de virtualización y paravirtualización decidí que el Sistema Operativo a utilizar sea Windows XP Profesional SP3. Ya que éste consume poca memoria, es estable y es confiable. La decisión de elegir este sistema operativo y no uno de software libre, como podría ser Ubuntu, está principalmente tomada en base a la memoria que consume cada uno de éstos, por ejemplo el Windows XP finalizada su instalación completa ronda los 350 MB., en cambio las últimas versiones de Ubuntu consumen aproximadamente unos 700 MB en su instalación inicial.

Para la virtualización el software a utilizar es VMWare Workstation 10.0.3 que es una de las últimas versiones y soporta sistemas operativos de los más recientes.

En cuanto para la paravirtualización fue utilizado para las pruebas piloto el software de Citrix XenServer Free 6.2.

3.4 Diseño del sistema.

En la figura 16 se observa el diseño a lograr con el Servidor Host y los Sistemas Operativos a ser virtualizados y paravirtualizados con sus respectivos software de aplicaciones.

Figura 16: Modelo del sistema Virtualizado.

Posteriormente para realizar los diferentes casos de usos, se utilizará una PC para la utilización de los distintos servicios que ofrecen los sistemas virtualizados. Como se observa en la figura 17.

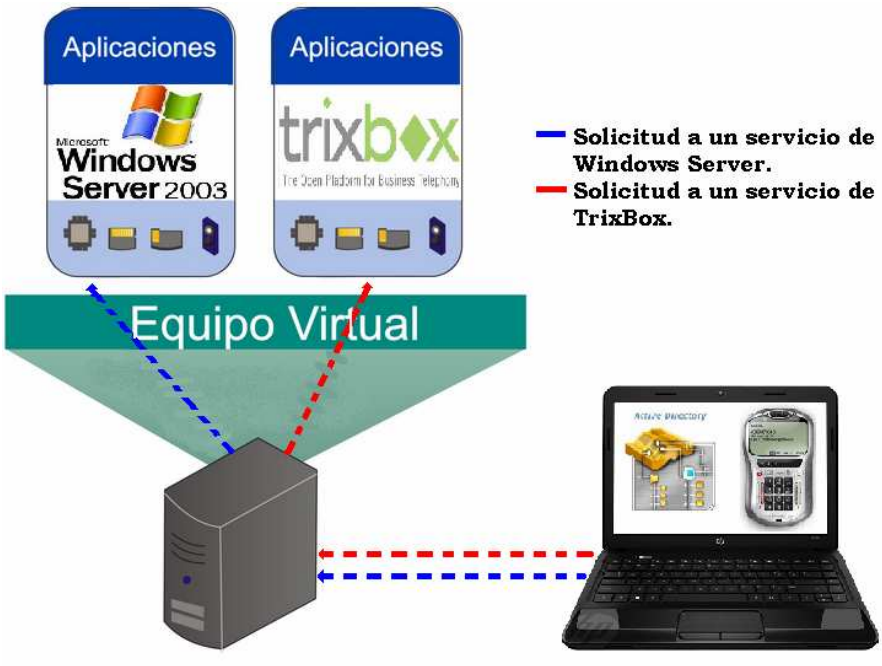

Figura 17: Acceso a los servicios virtualizados desde un equipo remoto.

3.5 Implementación sobre Virtualización.

3.5.1 Creando una Máquina Virtual en VMWare Workstation.

Una vez instalado el sistema operativo Windows XP con sus services packs, y habiendo instalado el software de virtualización VMWare estamos listos para instalar nuestro primer sistema a virtualizar. En la figura 18 observamos la pantalla de inicio del VMWare Workstation 10.

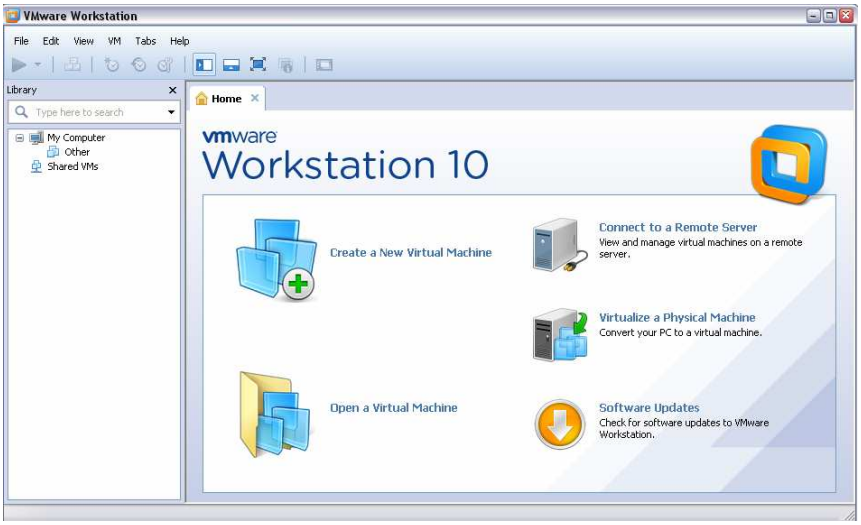

Figura 18: Pantalla inicio VMWare Workstation 10.

Para comenzar debemos Crear Nueva Máquina Virtual, donde aparecerá el cuadro de la figura 19, y elegimos personalizarla para poder elegir el hardware que necesitaremos para esta.

Figura 19: Creación de máquina virtual personalizada.

A continuación nos solicita saber si el sistema operativo a instalar será desde la lectora de CD’s o desde una imagen .ISO o instalarlo después. Cómo se observa en la figura 20.

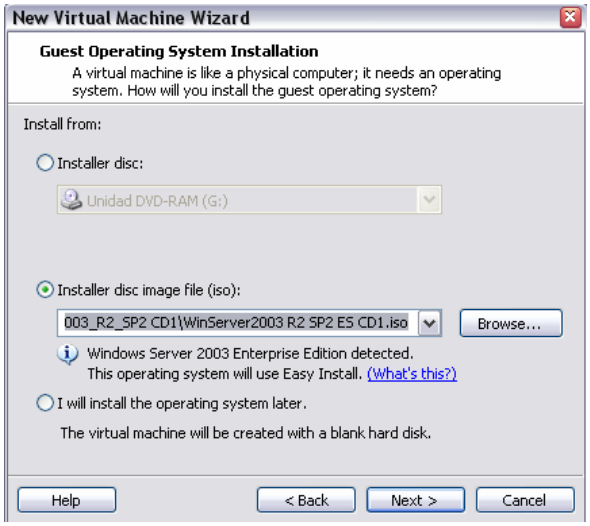

Figura 20: Ubicación de la fuente del sistema operativo.

Posteriormente, en la figura 21, nos solicita que especifiquemos la cantidad de Procesadores y Núcleos que va a utilizar ésta. Con lo cual hay que tener mucho cuidado con cuantos elegir ya que debemos saber con cuantos cuenta nuestro Servidor Host.

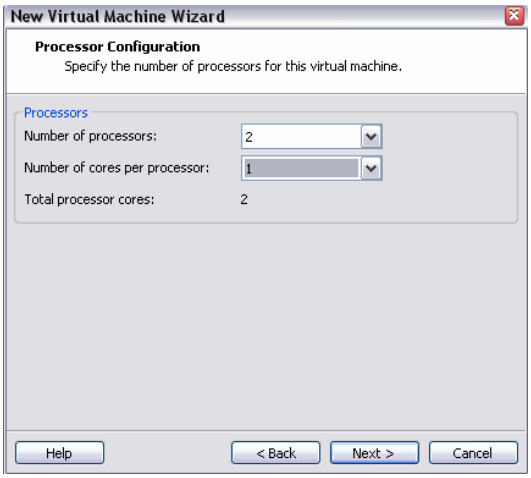

Figura 21: Procesadores y núcleos a asignar a la MV.

En la figura 22 nos solicita la cantidad de memoria RAM a asignar a ésta máquina virtual. Donde a cada una en este proyecto será asignado 1 GB de memoria RAM.

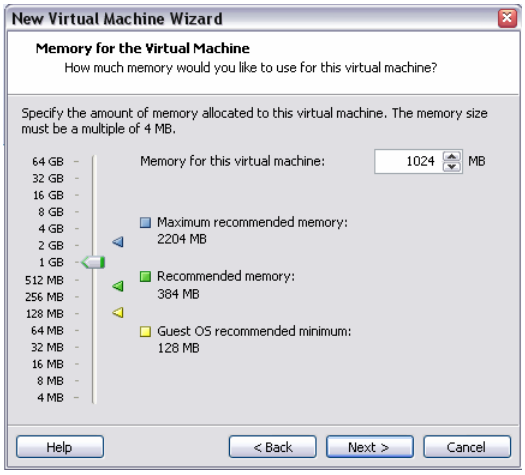

Figura 22: Cantidad de Memoria RAM para la MV.

A continuación, en la figura 23, nos solicita el Tipo de Red con que se conecta la máquina virtual a Internet, donde elegiremos por NAT (Traducción de direcciones de red). En éste punto hay que tener cuidado con la elección, ya que si necesitamos que el/los servidores tengan una dirección IP fija será necesario elegirlo en este paso.

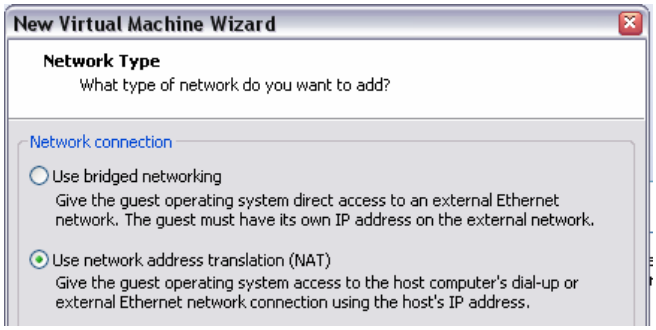

Figura 23: Elección del tipo de red.

Luego nos solicita el tipo de controlador de E/S SCSI, donde dejamos el recomendado por el sistema, que es LSI Lógico. También nos solicita el tipo de disco que queremos utilizar. Donde el sistema por defecto recomienda SCSI, por lo que utilizaremos éste.

A continuación, en la figura 24, elegimos el tamaño que va a tener el disco duro, como solo vamos a instalar el sistema operativo y pocas aplicaciones no es necesario un gran tamaño. Nos

permite, si queremos, reservar el espacio en el disco para no tener problemas posteriormente y si queremos que se guarde todo en un único archivo en múltiples por si necesitamos trasladar estos archivos.

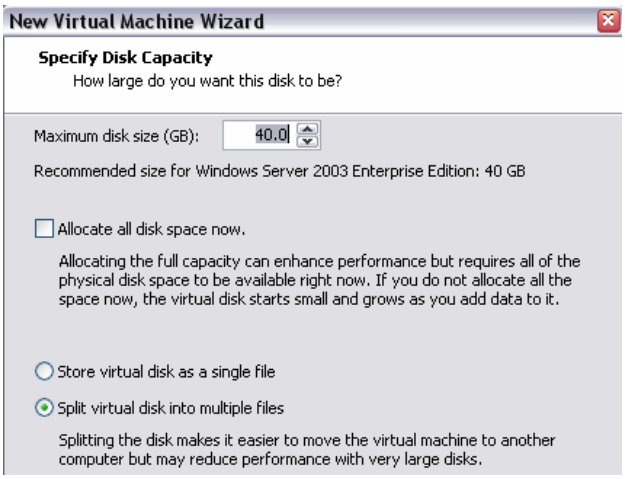

Figura 24: Reserva del tamaño del Disco Duro a utilizar.

Finalmente nos pide un nombre para guardar el archivo y nos presenta todo el informe final del sistema a instalar, figura 25. Y nos permite volver a modificar el hardware por si querríamos quitar o agregar algún dispositivo (USB, Impresora, Placa de Sonido, etc.).

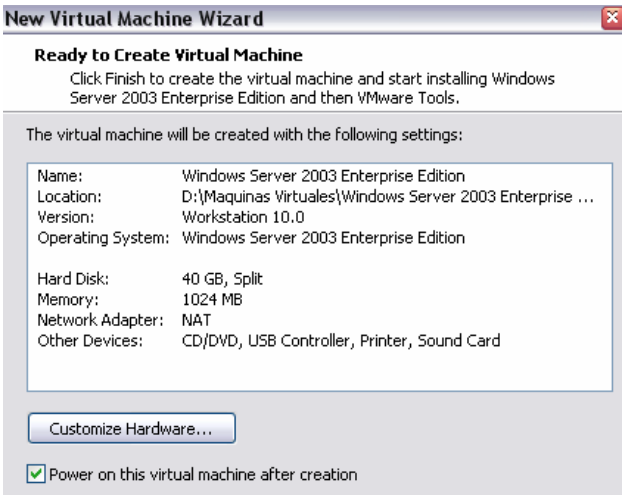

Figura 25: Informe final de la MV a crear.

Una vez finalizado el proceso de la creación de la MV, procedemos a encender la máquina virtual e instalar el sistema operativo Windows Server 2003.

3.5.2 Instalando Windows Server 2003.

Al encender la máquina virtual y bootear del CD de instalación del sistema, se procede con la instalación típica del sistema operativo, como se observa en la figura 26.

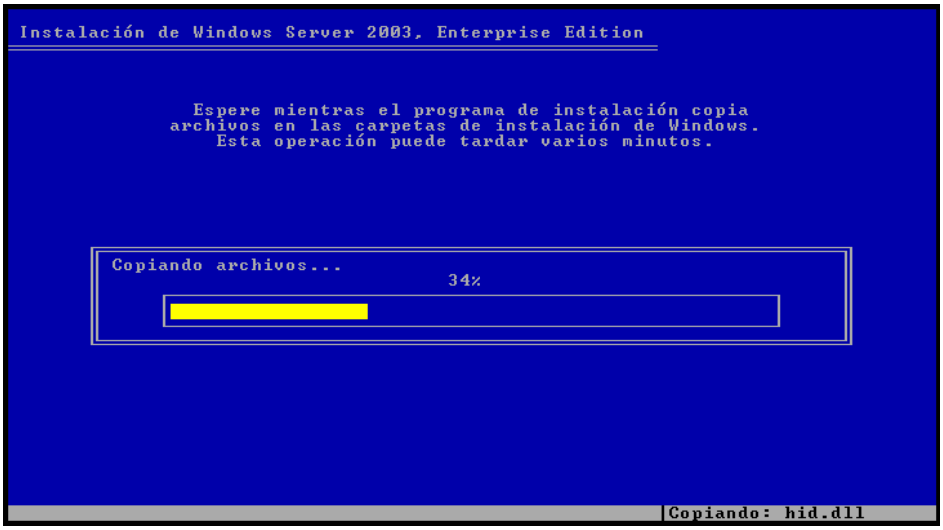

Figura 26: Instalación del sistema operativo.

Luego de seguir con todos los pasos necesarios para la instalación del sistema operativo, arranca nuestro Windows Server 2003.Y ya contamos con nuestro servidor funcionando, como podemos observar en la figura 27, y listo para configurar sus funciones, las cuales serán modificadas posteriormente.

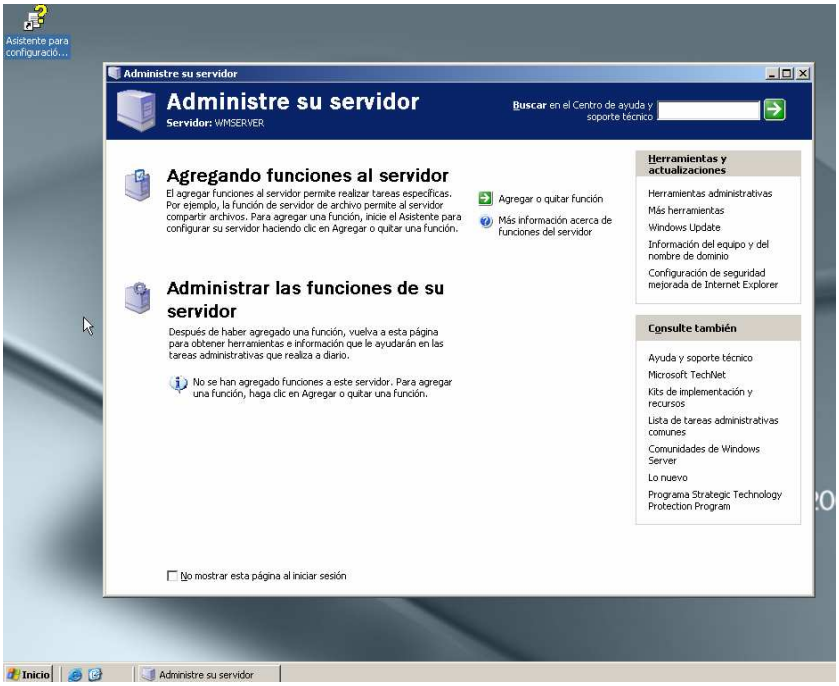

Figura 27: Escritorio de Windows Server 2003.

3.5.3 Instalando TrixBox CE.

En primer lugar se procede con la creación de la maquina virtual para este sistema operativo considerando las mismas características de hardware que para Windows Server. Una vez finalizado, estamos listos para comenzar con la instalación del sistema TrixBox. En la figura 28, observamos la MV TrixBoxCE ya creada con el mismo tipo de hardware que se utilizo para Windows Server 2003, excepto que se le asignó menos disco duro, y se encuentra lista para ser instalado el sistema.

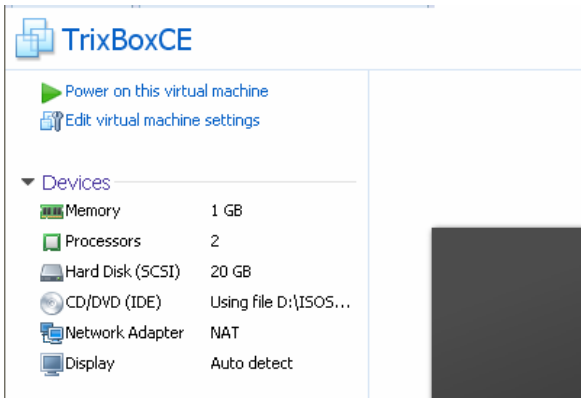

Figura 28: MV de TrixBox con sus especificaciones y lista para arrancar.

En la figura 29, observamos el booteo del sistema Trixbox para comenzar con su instalación. Donde informa la versión, como comenzar con la instalación y diferentes opciones de configuración.

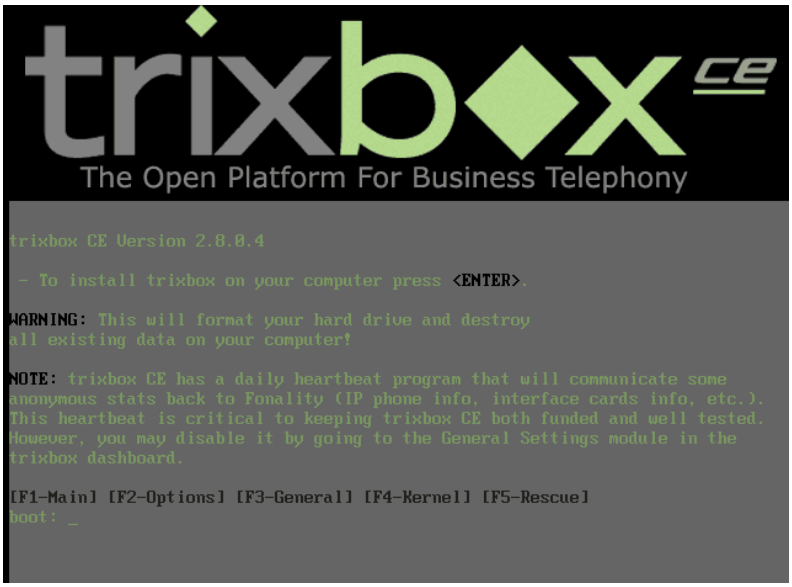

Figura 29: Pantalla inicial del sistema Trixbox.

Una de las ventajas de estas distribuciones es que instalan automáticamente todas las librerías, paquetes y el software necesario que se va a utilizar para las transferencias de llamadas de VoIP, como se observa en la figura 30.

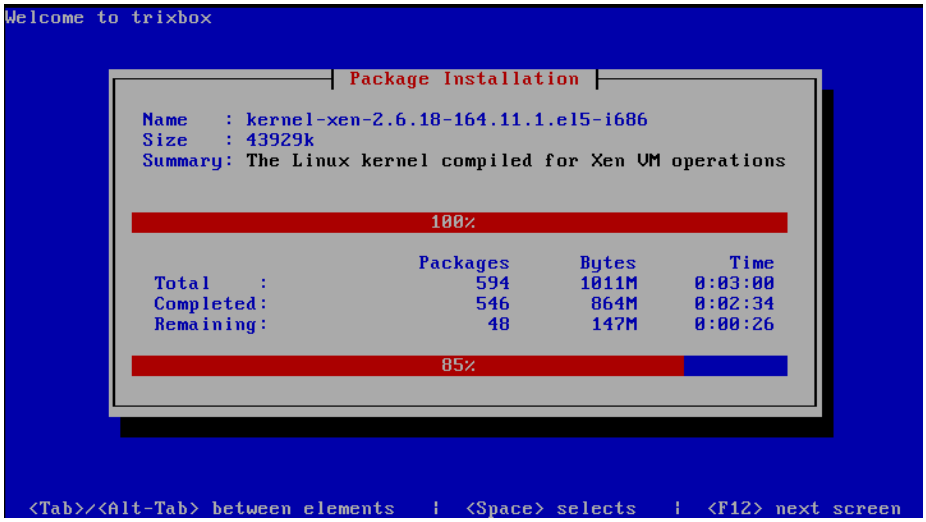

Figura 30: Instalación del sistema Trixbox.

Una vez finalizada la instalación, el servidor está listo para ser configurado, como muestra la figura 31, para poder acceder a esto y a todas sus funciones debemos ingresar al servidor vía Web.

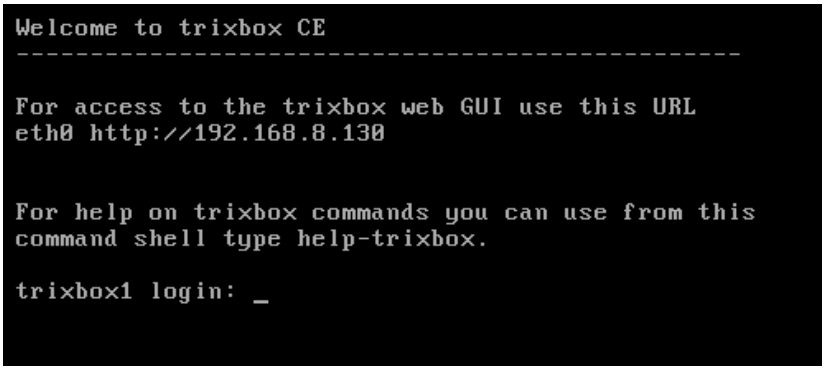

Figura 31: Servidor Trixbox instalado y listo para ser accedido vía Web.

Por lo que desde nuestra maquina virtual con Windows Server 2003 podremos acceder a la maquina virtual que corre TrixBox, como observamos en la figura 32.

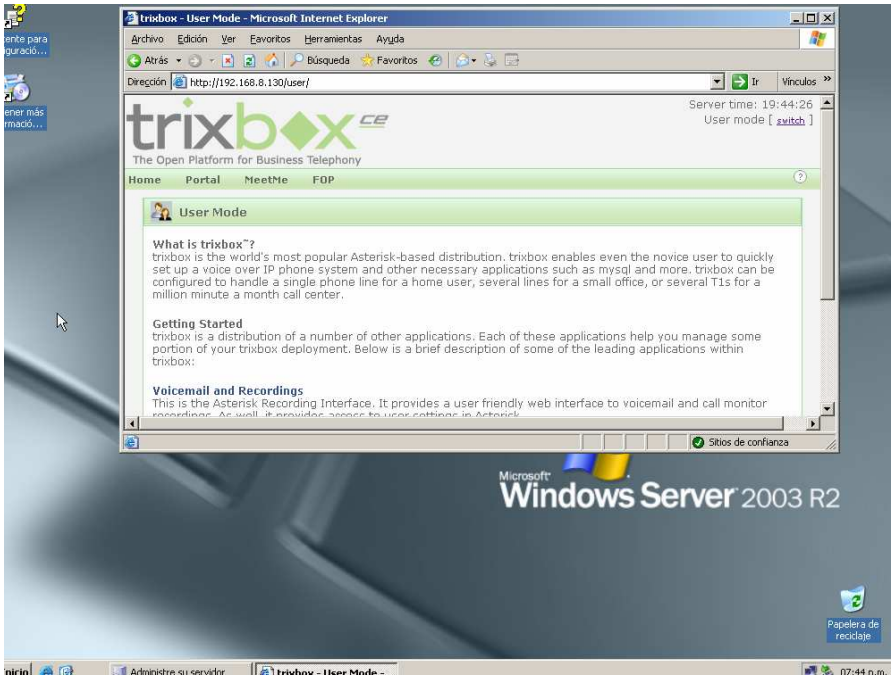

Figura 32: Acceso a Trixbox desde Windows Server 2003.

3.5.4 Configurando servicios en Windows Server 2003.

Este servidor ofrece una amplia variedad de funciones como ser: DNS, Servidor DHCP, servidor de archivos, de impresión, etc. Uno de los más importantes que brinda es el de Controlador de Dominio o Active Directory, éste servicio permite crear objetos tales como usuarios, equipos o grupos, con el objetivo de administrar los inicios de sesión en los equipos conectados a la red, así como también la administración de políticas en toda la red.

Su estructura jerárquica permite mantener una serie de objetos relacionados con componentes de una red, como usuarios, grupos de usuarios, permisos y asignación de recursos y políticas de acceso.

Como se muestra en la figura 33, procedemos a configurar Active Directory en nuestro servidor.

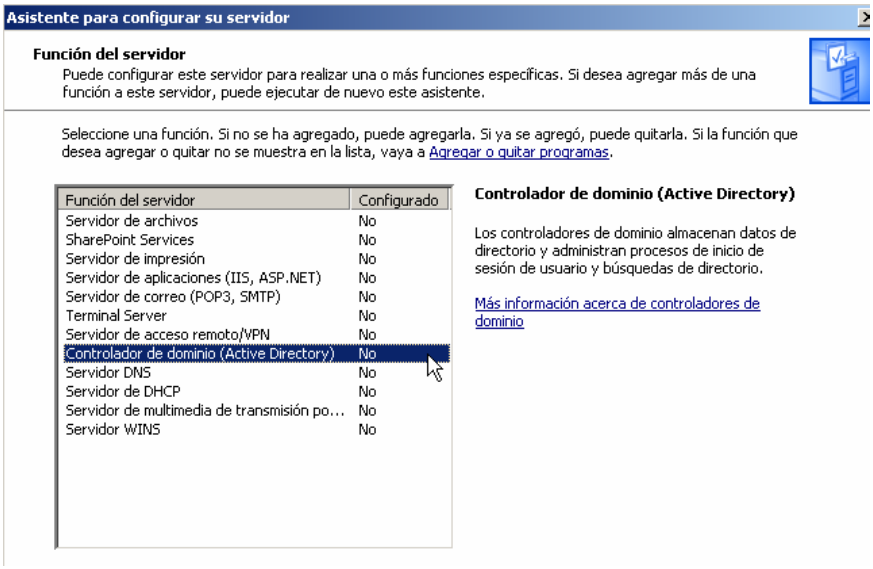

Figura 33: Elección del servicio a instalar en el servidor.

Comenzamos creando un nuevo Dominio para nuestra red. Cómo este será nuestro Dominio principal en la empresa, lo llamaremos Bosque. Como se observa en la figura 34.

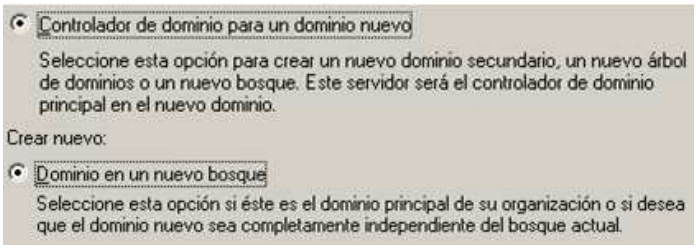

Figura 34: Creación de un Dominio, en un nuevo Bosque.

Para la creación del dominio es necesario crear un DNS (Domain Name System), que será el nombre con el cual identificamos al dominio en nuestra red. Como se observa en la figura 35.

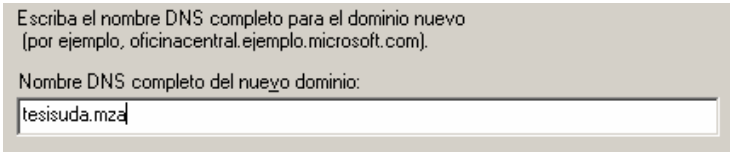

Figura 35: Asignación del nombre al servidor DNS.

Posteriormente debemos especificar el nombre a utilizar por NetBIOS, esto permite a las aplicaciones comunicarse en este dominio, como se describe en la figura 35.

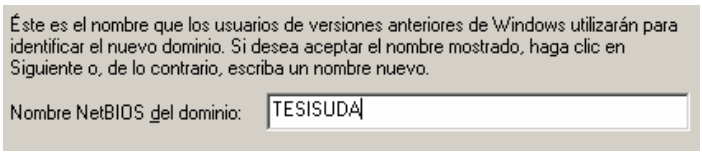

Figura 35: Asignación del nombre en NetBIOS.

Posteriormente la instalación continúa solicitando información sobre donde almacenar la configuración y luego de unos minutos el asistente finaliza y ya contamos con nuestro controlador de dominio, como observamos en la figura 36.

Figura 36: Active Directory instalado con éxito.

3.5.4.1 Creación de Grupos y Usuarios en Active Directory.

Una vez instalado el controlador de dominio, procedemos a crear grupos y usuarios con sus respectivas contraseñas y permisos para poder dar uso a esta aplicación lo cual es uno de los objetivos para poder visualizar el corrector funcionamiento del servidor y poder analizar su comportamiento.

En la figura 37, observamos como crear en Active Directory un nuevo grupo al que van a pertenecer los usuarios que crearemos posteriormente. El grupo será Prueba1.

Figura 37: Creación de Grupo en Active Directory.

Posteriormente se deben crear los usuarios que van a pertenecer al grupo. Los que vamos a utilizar son Usuario1 y Usuario2, como observamos en la figura 38.

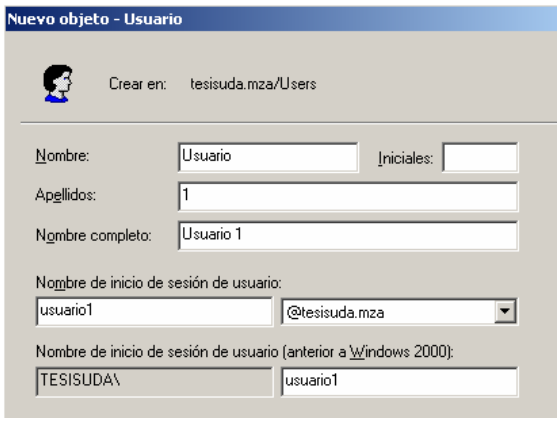

Figura 38: Creación de Usuarios en Active Directory.

Una vez creados los Usuario1 y Usuario2 continuamos con el procedimiento de asignarlos como miembros del grupo Prueba1, como se observa en la figura 39.

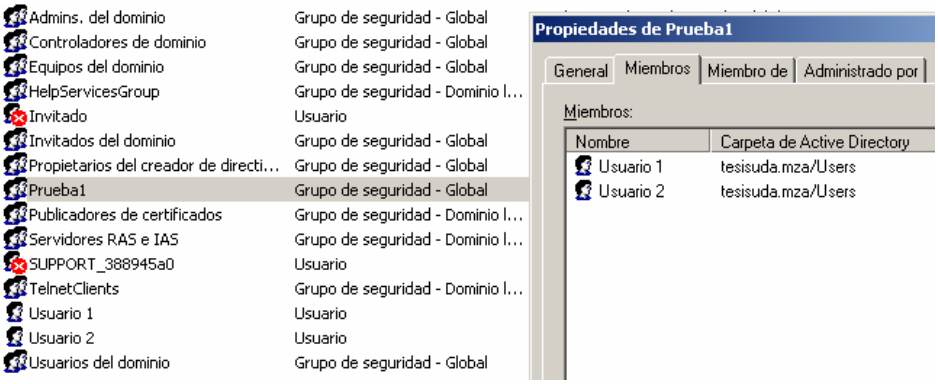

Figura 39: Asignación de Miembros a un Grupo.

3.5.4.2 Conectando una Terminal al Servidor.

A continuación para comenzar a utilizar este servicio en las terminales con Windows XP que se conectarán al servidor debemos configurar el dominio y el usuario que se utilizará. Como se puede apreciar en la figura 40.

Figura 40: Asociación de una Terminal a un Dominio.

Nos solicitará reiniciar el equipo Terminal para completar los cambios y al arrancar el Windows XP en el equipo Terminal, como muestra la figura 41, solicita el inicio de sesión con el usuario y la contraseña que se estableció.

Figura 41: Solicitud de inicio de sesión en un Dominio.

Finalmente estamos utilizando nuestra Terminal con Windows XP conectada al Dominio TESISUDA que está configurado en Windows Server 2003 virtualizado con el software VMWare Workstation. En la figura 42 observamos cómo sería el caso que estamos utilizando.

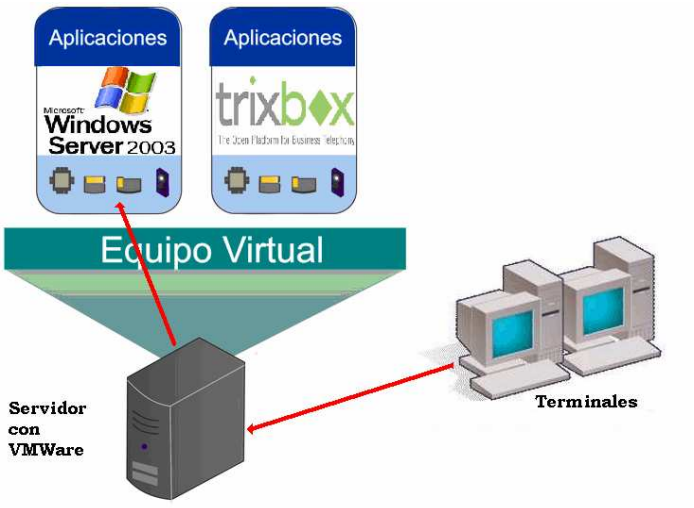

Figura 42: Solicitudes de aplicaciones desde las terminales hacia el servidor Windows Server 2003 Virtualizado.

3.5.5 Configurando TrixBox CE.

Para poder comenzar a utilizar este sistema de telefonía IP, lo primero que debemos hacer en TrixBox es crear los usuarios que se van a “conectar” a este servicio, para ello debemos crear extensiones SIP (Protocolo de Inicio de Sesiones), según nuestro cliente de VoIP que vallamos a

utilizar. En este también crearemos para nuestras pruebas el Usuario1 y Usuario2, en la figura 43 se observa la creación del primer usuario.

Figura 43: Creación de un usuario en Trixbox.

Cómo se dijo con anterioridad las facilidades de este sistema es que se instalan todos los paquetes y servicio para su uso, con solo configurar los usuarios ya estamos listos para utilizar las conexiones de telefonía IP que este brinda.

3.5.5.1 Configurando un cliente de VoIP.

Como se ve en la figura 44, procedemos con la configuración del cliente de VoIP, para ello utilizaremos el software X-Lite 4. Una vez instalado el software, configuramos la cuenta del usuario a conectar tal y como lo habíamos creado en el TrixBox.

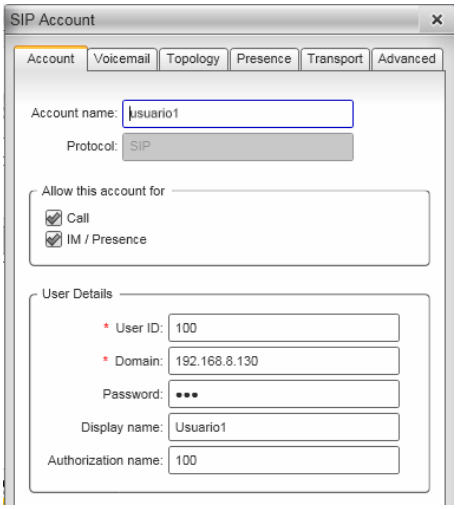

Figura 44: Creación de un cliente SIP en el software X-Lite 4.

Completados estos datos, el teléfono IP debería de conectarse y quedar en línea listo para realizar llamadas cómo observamos a continuación en la figura 45.

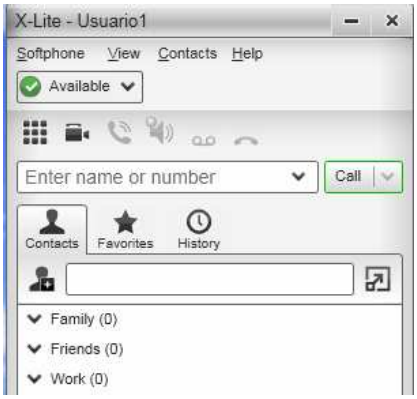

Figura 45: El usuario conectado al servidor Trixbox.

3.5.5.2 Realizando llamadas entre los clientes IP.

Ya con nuestros clientes de telefonía IP configurados, nos disponemos ha realizar llamadas entre ellos para comprobar un correcto funcionamiento del sistema. En la figura 46 observamos una llamada en proceso entre los Usuario1 y Usuario2.

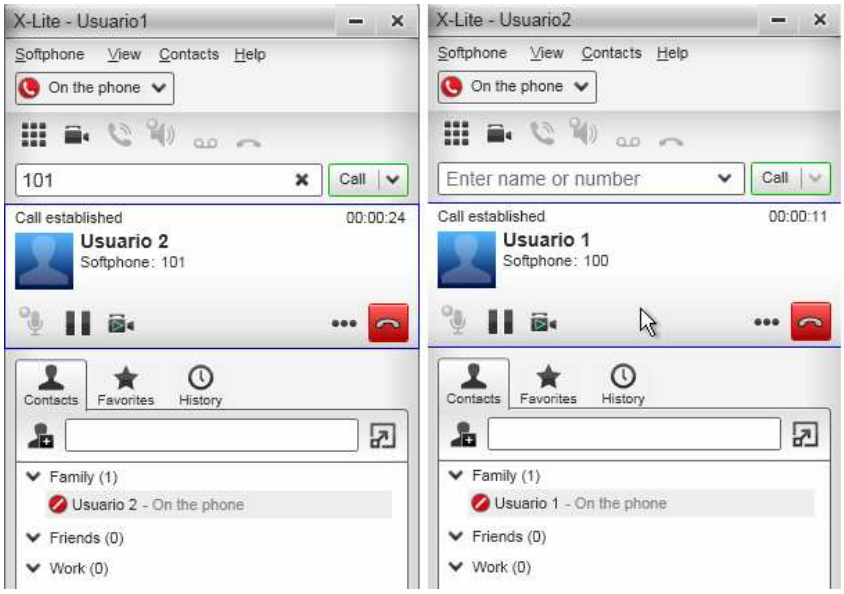

Figura 46: Llamada en proceso entre los dos usuarios creados.

Si observamos la figura 47, en el monitor del estado del servidor de TrixBox, podemos distinguir los dos equipos conectados.

Figura 47: Estado del servidor Trixbox con dos usuarios conectados.

El diagrama del caso de uso que se está realizando en este caso sería el que se observa a continuación en la figura 48.

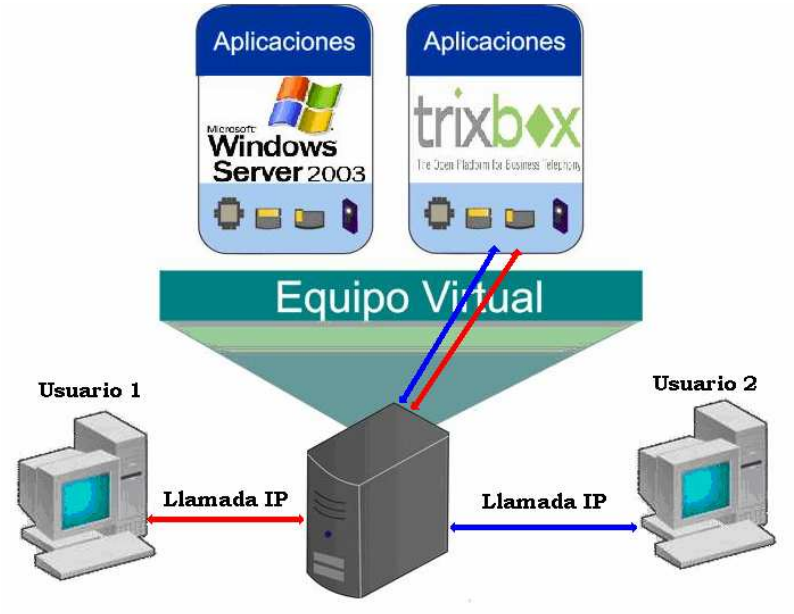

Figura 48: Procedimiento de la llamada de VoIP a través del servidor virtual.

3.6 Implementación sobre Paravirtualización.

3.6.1 Instalando XenServer.

En este paso debemos instalar Citrix XenServer Free Edition para convertir al equipo en un servidor de virtualización dedicado. Lo cual quiere decir que en este se almacenarán las máquinas virtuales, pero serán administradas desde otro equipo con el software XenCenter.

En la figura 49 observamos el proceso de bootear la PC con el CD y comenzar con la instalación del servidor.

Figura 49: Pantalla de booteo de Citrix XenServer.

En los pasos siguientes se solicitan datos, que al igual que en la instalación de cualquier sistema operativo estos son: idioma, configuración de teclado, y demás datos que no son relevantes, excepto por la configuración de la red, donde debemos de configurar una dirección IP en el rango en el que este configurada nuestra LAN, tanto como el DNS y el hostname.

Una vez completado todos estos datos, el sistema procede con la instalación como se observa en la figura 50.

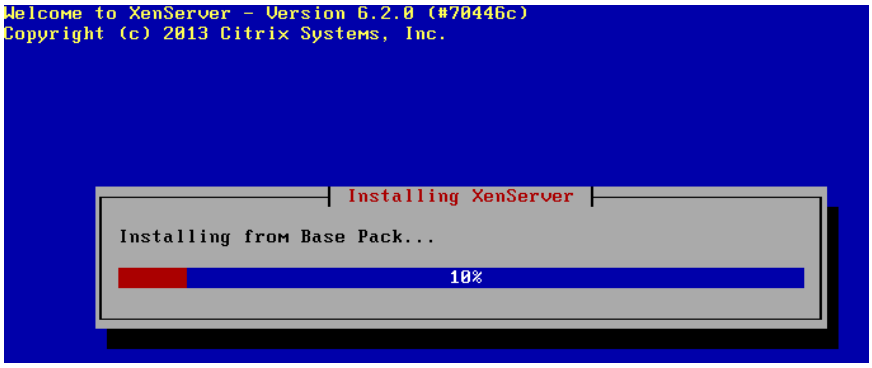

Figura 50: Instalación de XenServer en proceso.

Una vez instalado nuestro servidor con el XenServer, podemos acceder por consola y nos mostrará una ventana desde donde podremos configurar y administrar el servidor, como se aprecia en la figura 51.

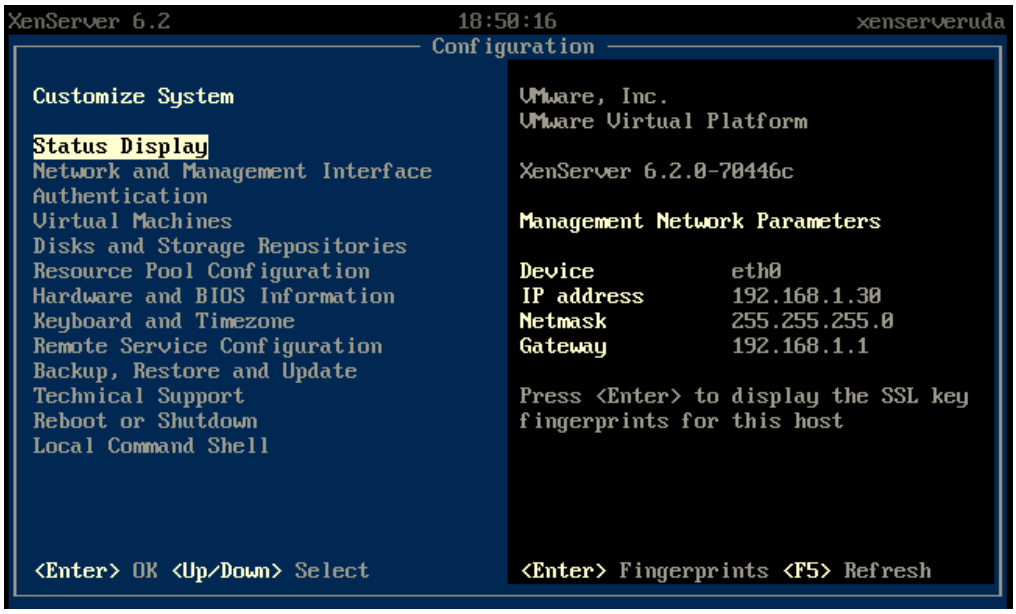

Figura 51: Pantalla de Configuración del servidor XenServer mediante consola.

3.6.2 Conectando XenCenter al servidor.

Una vez instalado el software cliente para poder acceder al servidor, podremos administrar nuestras máquinas virtuales a través de él. Para ello debemos conectar el cliente con el respectivo servidor, se debe especificar la dirección IP que asignamos a éste, como se observa en la figura 52.

Figura 52: Conectando el software XenCenter al servidor XenServer.

Una vez conectados nos da información sobre el uso del CPU y la memoria que está consumiendo en ese momento, como vemos en la figura 53.

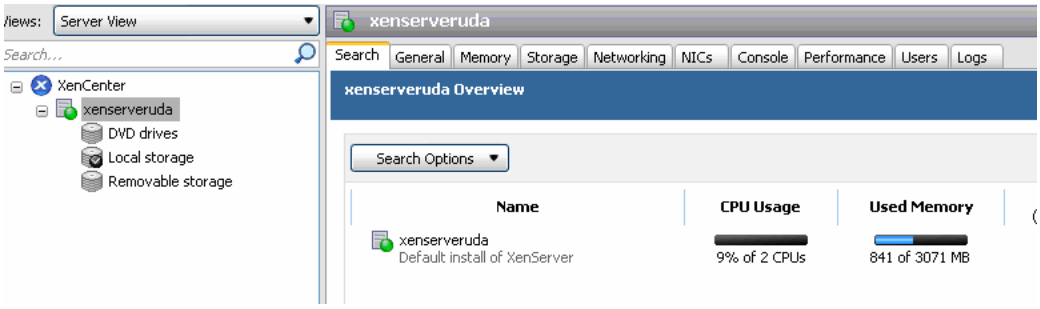

Figura 53: Estado del servidor y consumo de CPU y Memoria.

3.6.3 Creando una MV para Windows Server 2003 y Trixbox.

Para crear la nueva máquina virtual para Windows Server 2003 debemos seguir los siguientes pasos. Primero elegimos el sistema a instalar, como se observa en la figura 54.

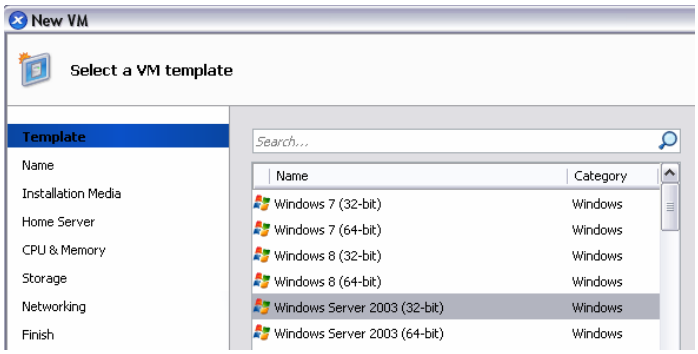

Figura 54: Elección del sistema operativo a instalar.

A continuación procedemos con el nombre y si queremos una descripción. Debemos configurar desde dónde se instalará, la cantidad de memoria y de núcleos a utilizar, como se ve en la figura 55.

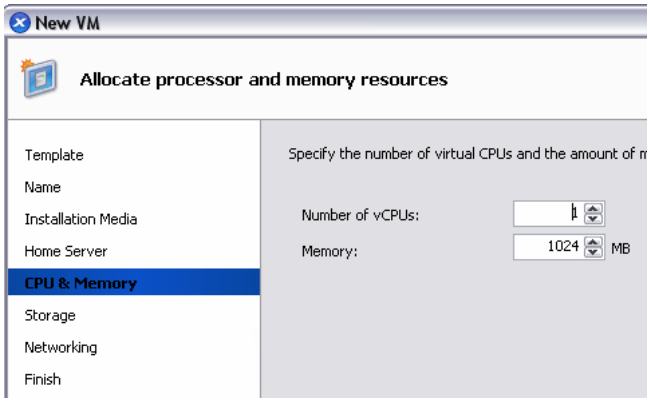

Figura 55: Asignación de CPUs y memoria.

Para el almacenamiento tenemos que crear una partición de disco virtual para almacenar este servidor, como vemos en la figura 56.

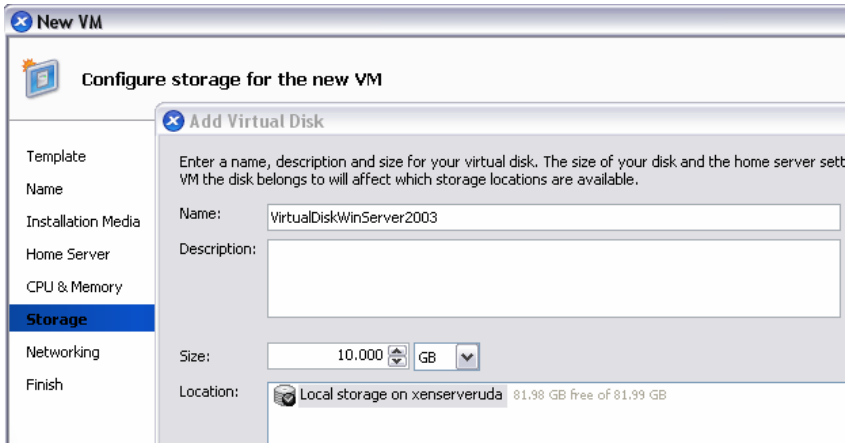

Figura 56: Asignación del disco y tamaño.

En la figura 57 observamos que para la configuración de la red lo recomendado es elegir que cree una red automática que generará una dirección MAC automática por el sistema.

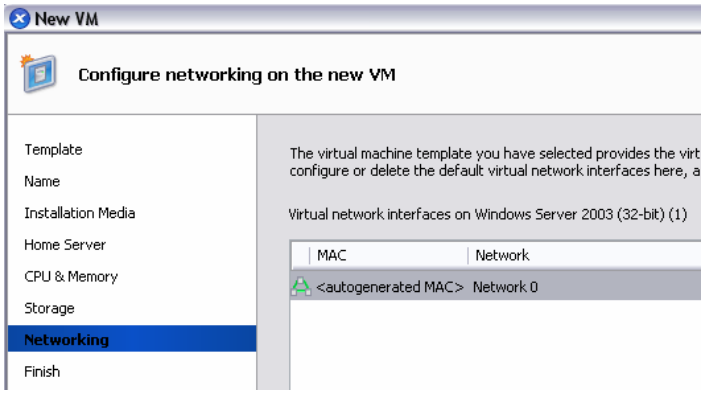

Figura 58: Elección de red.

Y damos por finalizada la creación de ésta MV para proceder con la instalación del sistema operativo.

Se procede con los mismos pasos pero ahora para la creación de la MV que hospedará al sistema TrixBox. Una vez creadas ambas MV, podemos observar tanto desde consola en el servidor dedicado como en el aplicativo que las máquinas quedaron listas, como se ve en las figuras 59 y 60.

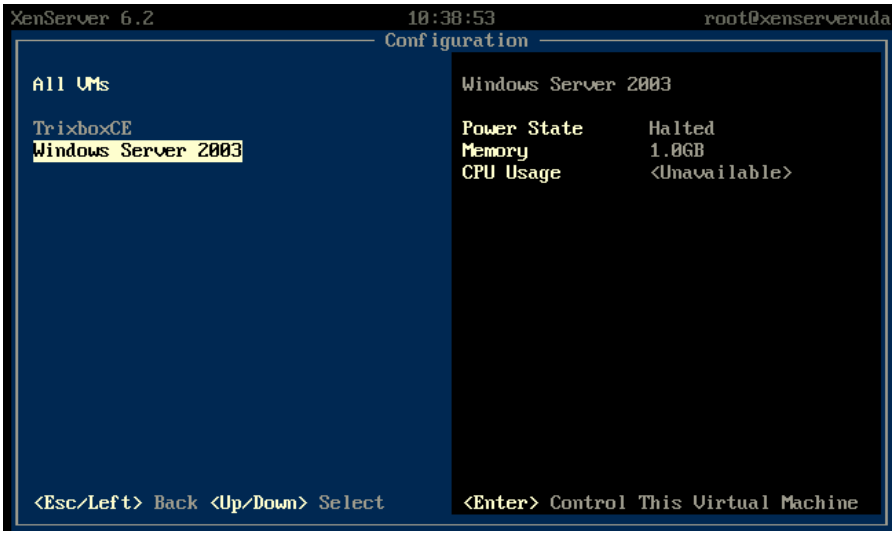

Figura 59: Estado de las MV a través de consola.

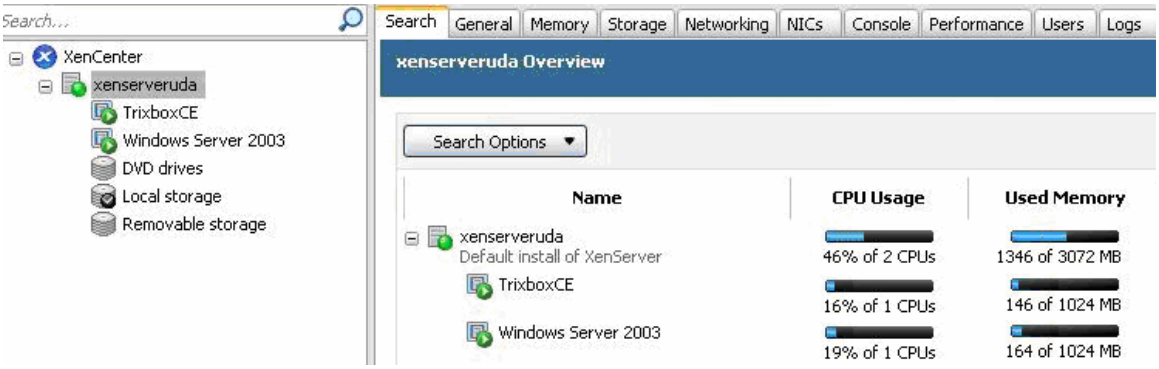

Figura 60: Estado de las MV a través del software cliente.

3.6.4 Instalando los sistemas operativos.

Los pasos para la instalación de ambos sistemas operativos son los mismos que los ya explicados con anterioridad, por lo que no entraremos en detalle sobre la instalación de éstos ya que no hay ningún punto a describir.

En la imagen 61 se observa el booteo del sistema Trixbox CE para el comienzo de su instalación:

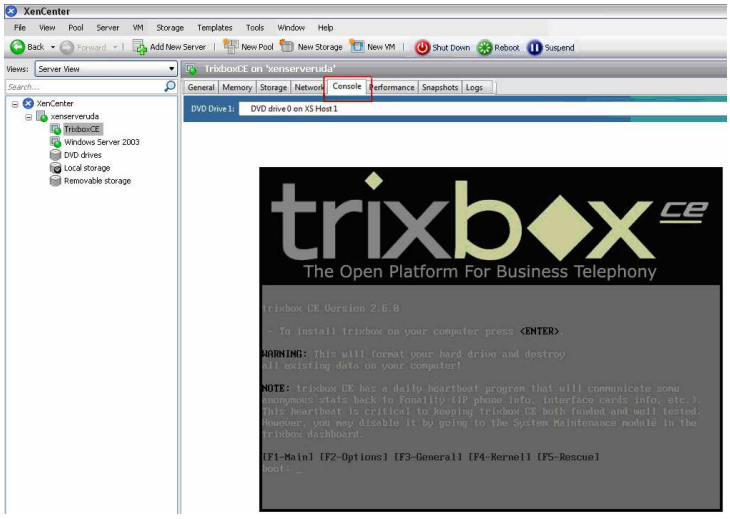

Figura 61: Instalación de Trixbox a través del cliente XenCenter.

A continuación el comienzo de la instalación de Windows Server 2003, como se ve en la figura 62.

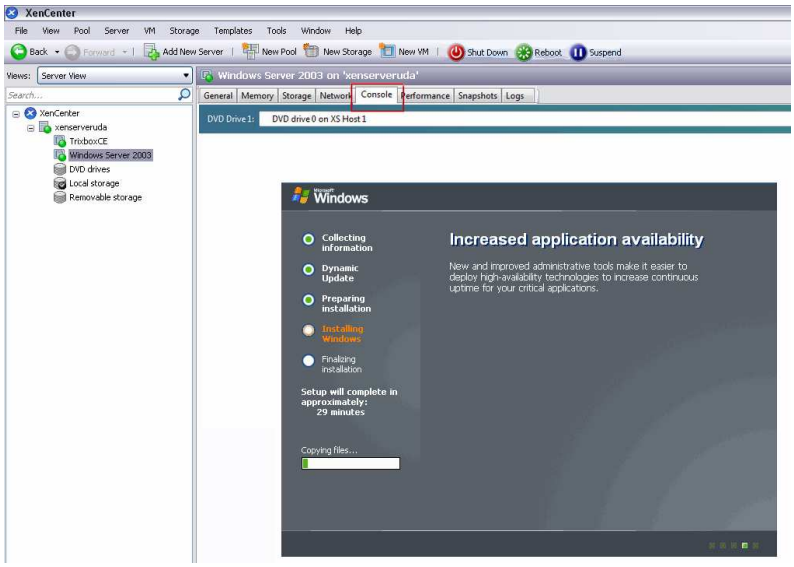

Figura 62: Instalación de Windows Server 2003 mediante el uso del cliente XenCenter.

Una vez instalados nuestros dos servidores, procedemos con la configuración de sus respectivos servicios, tal como lo hicimos con anterioridad en los sistemas virtualizados.

3.6.5 Probando los servicios paravirtualizados.

A continuación se procede a instalar en ambos servidores los servicios que brindarán, en Windows Server 2003 el controlador de dominios y en TrixBox creamos las extensiones para conectar los teléfonos IP.

Para éstos no entraremos en detalles sobre la instalación de cada uno, simplemente diremos que los sistemas quedaron funcionando correctamente tras realizar las configuraciones y pruebas pertinentes.

Una vez finalizadas las instalaciones y las pruebas de ambos sistemas, ya sean virtualizados o paravirtualizados, estamos listos para proceder con los análisis y comparaciones para determinar cual de los sistemas es conveniente utilizar a la hora de necesitar instalar alguno de estos.

Capítulo 4: Análisis y Pruebas de rendimiento.

4.1. Virtualización.

Para comenzar con nuestro análisis realizaremos la comparación de los sistemas que fueron instalados en la virtualización de nuestros servidores.

En la figura 63 se detallan las características y especificaciones del hardware pertenecientes al host anfitrión, las cuales se deben tener en cuenta al momento de la asignación de recursos para los servidores que serán Virtualizados.

DISPOSITIVO	CARACTERISTICAS		
	WinXP Sistema Anfitrión	Windows Server 2003 Virtualizado	TrixBox CE Virtualizado
Procesador	AMD Athlon 64 X2 5200+	AMD Athlon 64 X2 5200+	AMD Athlon 64 X2 5200+
Núcleos	2	2	2
Frecuencia	2600 MHz	2600 MHz	2600 MHz
Cache L1	64 KB por núcleo	64 KB por núcleo	64 KB por núcleo
Cache L2	1 MB por núcleo	1 MB por núcleo	1 MB por núcleo
Memoria RAM	3070 MB	1023 MB	1023 MB
Disco Duro	500 GB	40 GB	20 GB
Tarjeta de Red	NVIDIA nForce Networking	MT PRO/1000 de Intel	1394 Net Adapter

Figura 63: Características del host anfitrión con respectos a los sistemas virtualizados.

Estos datos fueron extraídos de los propios sistemas, por lo que podemos apreciar que prácticamente las características son las mismas en lo que respecta al procesador y frecuencias, porque esto es tomado de la placa madre, exceptuando la memoria, el espacio en disco que les fue asignado, y los respectivos drivers de red que cada sistema utiliza para controlar este periférico.

4.2 Uso del CPU.

El manejo del CPU se realiza bajo un componente llamado “scheduler” el cual se encarga de realizar la asignación de cada núcleo en un intervalo aproximado de 20 msg. Si uno toma datos de performance sobre los sistemas actuales, el promedio no supera el 15 % de uso de CPU.

A continuación se presenta el análisis realizado al host anfitrión para verificar la utilización del CPU mediante el software EVEREST, se obtuvieron los resultados que se observan en la figura 64. Los datos extraídos corresponden a distintos estados que atravesó el host durante las instancias de arranque de los distintos sistemas virtualizados, como se detallan en las figuras 64 y 65.

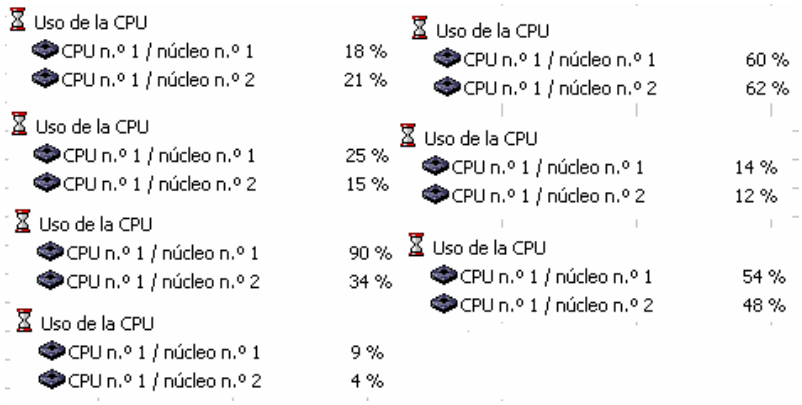

Figura 64: Uso del CPU en diferentes momento del host anfitrión.

	USO DEL CPU	Host Anfitrión	
	Estados del Equipo	Núcleo N° 1	Núcleo N° 2
1	Arranque del S.O.	45%	53%
2	Sistema en Ejecucion sin Aplicacióne	2%	1%
3	Arranque de Vmware	25%	15%
4	Posterior estabilidad del sistema	3%	6%
5	Arrnaque de Windows Server 2003 Vi	90%	34%
6	Posterior estabilidad del sistema	9%	4%
7	Arranque de TrixBox Virtual	60%	62%
8	Posterior estabilidad de los sistemas	14%	12%
9	Acceso a servicios de los sistemas Vi	54%	48%

Figura 65: Detalle de los estados y usos del CPU.

De la extracción de estos datos observamos que el mayor uso generado por los núcleos que componen el CPU es generado principalmente al momento de arranque de los equipos virtuales, y un muy bajo consumo de este recurso en los momentos en que no se generan peticiones a los

servidores. En la figura 66 se aprecian con mas claridad los picos generados al arrancar los sistemas.

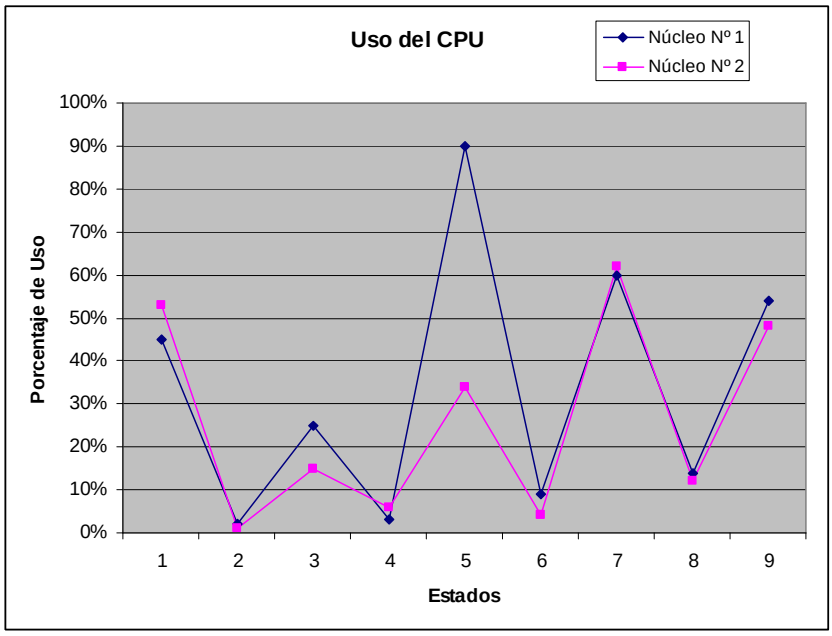

Figura 66: Estados del CPU.

En el caso planteado en esta tesina, como ya se explicó anteriormente, el host anfitrión utiliza un solo procesador con dos núcleos, con lo extraído, vemos que para montar dos máquinas virtuales, en los momentos más críticos estos trabajan a más del 50% cada uno. Por lo que si necesitásemos instalar más de dos equipos virtuales sería necesario contar con un procesador que cuente con una mayor cantidad de núcleos para alivianar el trabajo de estos.

4.2.1 Análisis del uso de memoria RAM.

En la tabla de la figura 67, observamos el uso de memoria del sistema anfitrión, en primer lugar con el sistema recién arrancando, luego ejecutando VMWare con Windows Server 2003, con TrixBox y luego con ambos sistemas virtualizados ejecutándose.

Memoria Física	CASOS			
	WinXP Sistema Anfitrión al arrancar	Con Windows Server 2003 Virtualizado	Con TrixBox CE Virtualizado	Con los dos Sistemas Virtualizados
TOTAL	3070 MB	3070 MB	3070 MB	3070 MB
USADA	556 MB	1235 MB	965 MB	1542 MB
LIBRE	2514 MB	1836 MB	2015 MB	1529 MB
USO	18%	40%	31%	51%

Figura 67: Uso de memoria en el host anfitrión en diferentes casos.

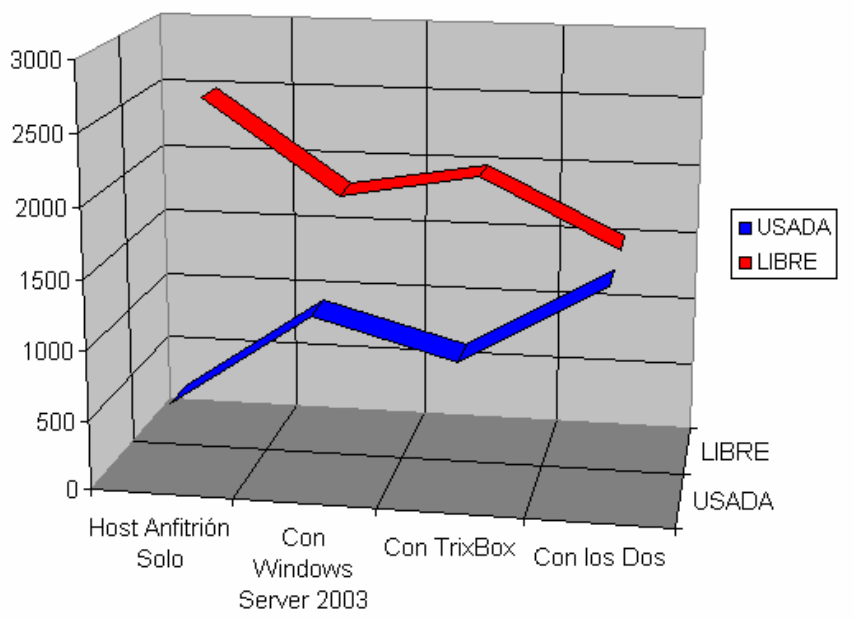

Figura 68: Gráfico del consumo de memoria.

Con estos datos podemos observar como el consumo de memoria RAM en el equipo anfitrión comienza a aumentar al virtualizar los servidores, vemos que solo con el sistema Trixbox el consumo de memoria es menor que con Windows Server 2003, ya que como se explicó con anterioridad esta distribución para llamadas IP es muy “liviana”. Y finalmente con los dos sistemas virtualizados, el consumo de la memoria del equipo anfitrión supera el 50% del consumo de la memoria. Cabe aclarar que estos datos fueron tomados sin estar realizando peticiones a estos servidores por lo que consideramos que el uso de la memoria debería de aumentar mucho más según los usuarios que se conecten a estos.

El software de virtualización a menudo incluye características para manejar la sobreasignación de la memoria. VMware ofrece compartición de memoria y capacidades de intercambio/englobamiento para compartir y reasignar de forma dinámica los recursos limitados de memoria. Sin embargo, con suficiente memoria física, se minimiza el efecto sobre el rendimiento de estas características.

Ahora procedemos con realizar una prueba de Benchmark a la memoria con el software EVEREST, en las mismas etapas que realizamos el test anterior. En la figura 69, se realiza primero con el sistema anfitrión recién arrancado observamos el siguiente resultado.

Figura 69: Estado de la memoria del host anfitrión con el equipo recién iniciado.

Una vez virtualizado el servidor de Windows Server 2003 observamos, en la figura 70, que el acceso a la memoria de escritura y de lectura a aumentado unos 100 MB/s. El tiempo de acceso continúa con valores similares a los anteriores. Pero nos encontramos con una Latencia que ha aumentado unos nano/segundos, con lo cual, si consideramos el poco uso que tenemos en la red, en una empresa con grandes cantidades de solicitudes este factor se vería afectado desfavorablemente y generaría demoras en las respuestas de los servicios.

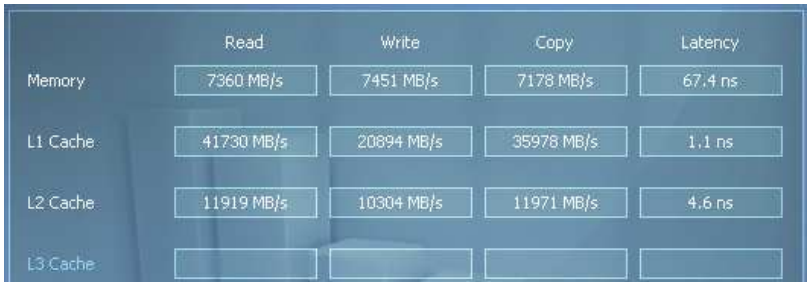

Figura 70: Estado de la memoria con el primer servidor en funcionamiento.

Se procede con la virtualización del sistema de TrixBox, quedando así nuestros dos servidores encendidos, y realizamos nuevamente el test de acceso a memoria en el anfitrión y el resultado que obtenemos es un aumento en los accesos a lectura y escritura, y un no tan notado aumento en el tiempo de Latencia, como se puede ver en la figura 71.

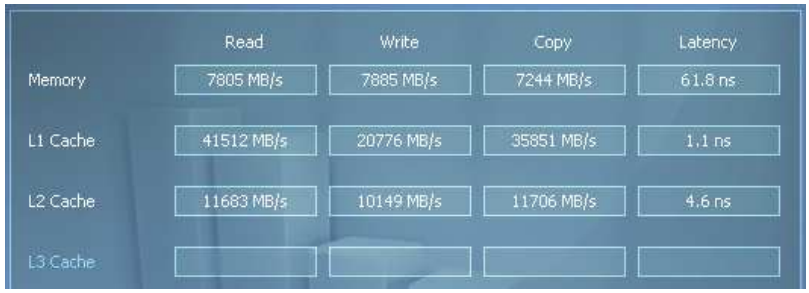

Figura 71: Estado de la memoria con los dos sistemas virtualizados.

Una motivación clave para la virtualización, es mejorar la utilización de recursos de hardware, manteniendo una calidad del servicio. Esto se logra con una gestión eficiente de los recursos, hay que tener en cuenta que la mayoría de los recursos físicos tales como el núcleo del procesador, dispositivos E/S, se comparten con las maquinas virtuales y se pueden programar de tal manera de que se utilicen mediante prioridades. Pero para la asignación de memoria es mucho más difícil, esto es puesto que distintas aplicaciones usan de variadas forma la memoria, incluso “asignándose” durante la ejecución. Por lo tanto se necesita asignar/ajustar dinámicamente la memoria para cada máquina virtual. Es en donde aparece el concepto de MEB (Balanceador de memoria) que controla dinámicamente el uso de la memoria, además utiliza un modelo de predicción para el uso y necesidades de esta memoria para luego reasignar esta memoria al host.

4.2.2 Prueba de estabilidad del sistema.

A continuación se realizan pruebas de estabilidad al sistema en diferentes puntos; cuando iniciamos el software de virtualización VMWare, al arrancar Windows Server 2003, al iniciar TrixBox y mientras se realizaron pruebas a los servidores. Estos datos fueron capturados por una Test de Estabilidad del Sistema perteneciente al software EVEREST.

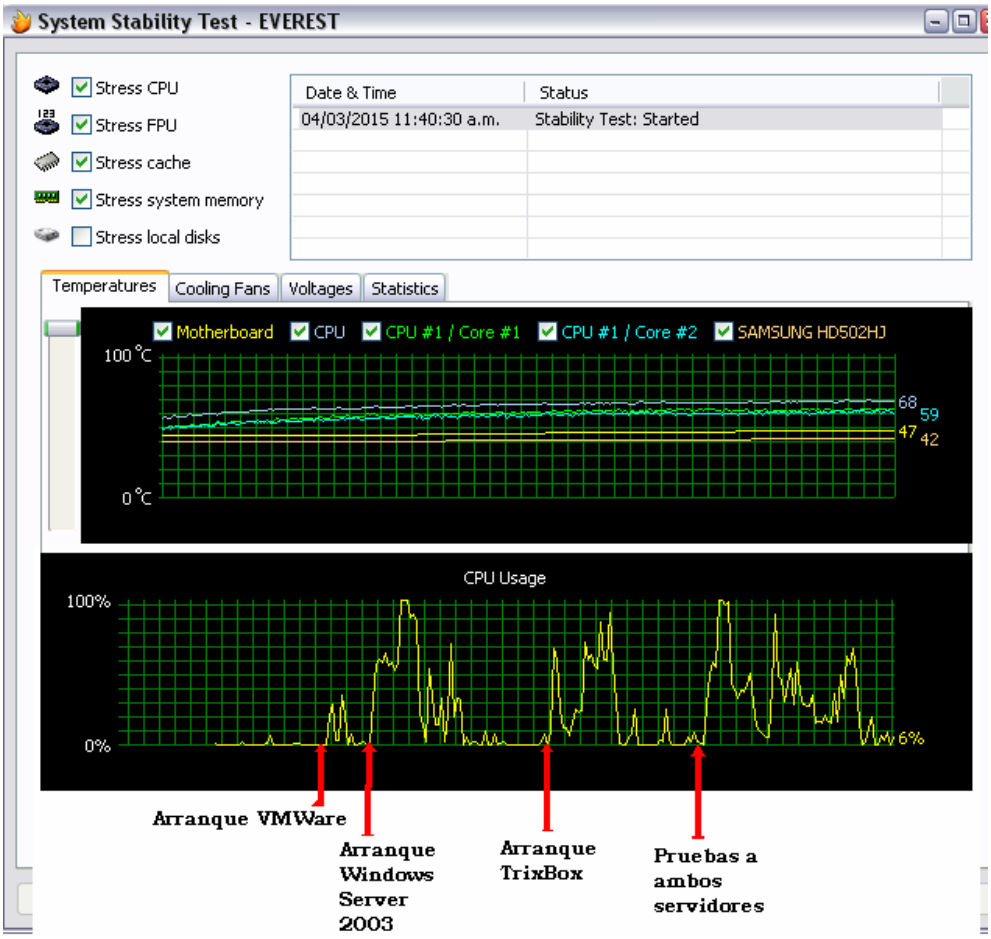

Figura 72: Uso del CPU en el proceso de arranque de los sistemas virtualizados.

Como podemos observar en la figura 72, los picos en el uso del CPU del host anfitrión durante las etapas de arranques de los sistemas y en los momentos en que se solicita un mayor uso de sus respectivos servicios brindados. Una vez que ambos sistemas están en ejecución y se comienzan a utilizar el uso del CPU supera por momento el 50% de su uso.

4.3 Pruebas sobre la Paravirtualización.

Tal como hicimos anteriormente realizaremos una comparación de las especificaciones del hardware que se asignó a cada servidor Paravirtualizado con respecto al host anfitrión. En la figura 73 se observan las características del servidor "dedicado" para XenServer y los sistemas paravirtualizados.

DISPOSITIVO	CARACTERISTICAS		
	XenServer Sistema Anfitrión	Windows Server 2003	Trixbox CE
Procesador	AMD Athlon 64 X2 5200+	AMD Athlon 64 X2 5200+	AMD Athlon 64 X2 5200+
Núcleos	2	2	2
Frecuencia	2600 Mhz	2600 Mhz	2600 Mhz
Cache L1	64 KB por núcleo	64 KB por núcleo	64 KB por núcleo
Cache L2	1 MB por núcleo	1 MB por núcleo	1 MB por núcleo
Memoria RAM	3 GB	1 GB	1GB
Disco Duro	500 GB	40 GB	20 GB
Red	PCnet32 LANCE	MT Pro /1000 Intel	1394 Net Adapter

Figura 73: Características del servidor dedicado y los sistemas paravirtualizados.

Con los datos extraídos de la configuración de cada uno de los sistemas, observamos que las características, tanto del anfitrión como de los equipos paravirtualizados detectan el mismo procesador con su frecuencia y asignaciones de caches. Los tamaños de memoria son los asignados por el administrador al igual que el tamaño del disco. Y se distinguen el tipo de controlador (drivers) que utiliza cada uno de estos para el control de la red.

4.3.1 Uso de memoria RAM.

Gracias al software de XenCenter podemos observar en una de sus pestañas la utilización de memoria del sistema anfitrión y los sistemas paravirtualizados. En la figura 74, observamos el uso de la memoria.

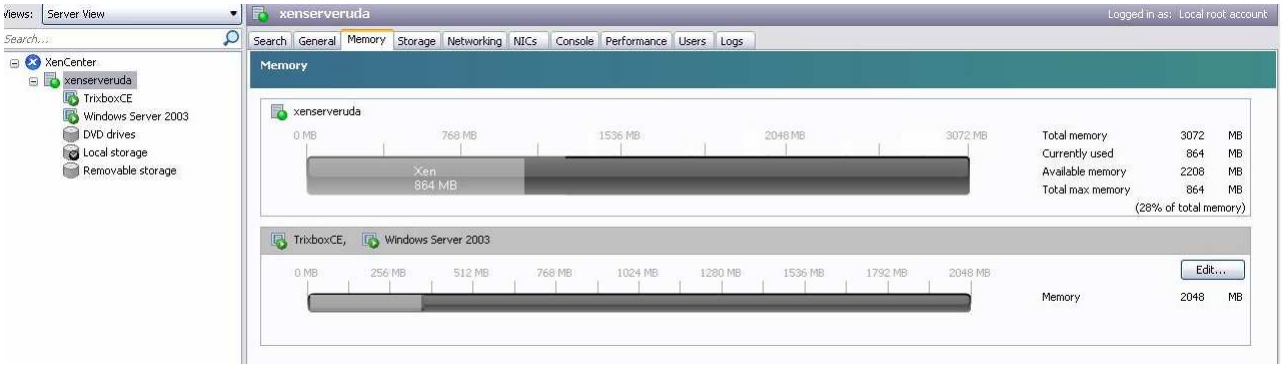

Figura 74: Uso de la memoria RAM mediante el cliente XenCenter.

Se observa que entre los dos sistemas paravirtualizados están consumiendo un aproximado de 350 MB. Y el sistema anfitrión posee un consumo promedio de los 900MB de un total de 3 GB disponibles. De estos datos surge la figura 75 donde se pueden observar que el consumo de memoria de este método es mucho menor al observado con los sistemas virtuales.

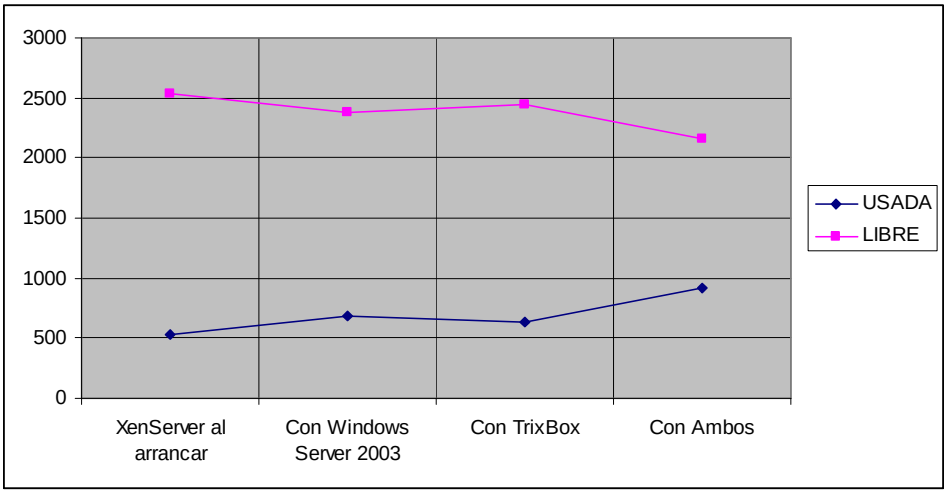

Figura 75: Gráfico de consumo de memoria.

4.3.2 Usos del CPU, Disco, Red y Memoria.

También podemos visualizar en detalle el uso de cada uno de estos elementos, en la imagen 76, observamos el uso del CPU, de la Memoria, del Disco y de las interfaces de Red. También nos indica la dirección IP de cada servidor y el tiempo que ha estado en ejecución. Teniendo en cuenta

estos datos y los extraídos con anterioridad para los sistemas virtualizados. Observamos como el método de paravirtualización administra de mejor manera el uso del CPU. Donde una vez estabilizados los sistemas con la virtualización el uso promedio del CPU ronde los 40% mientras que con la paravirtualización es aproximadamente un 25 %.

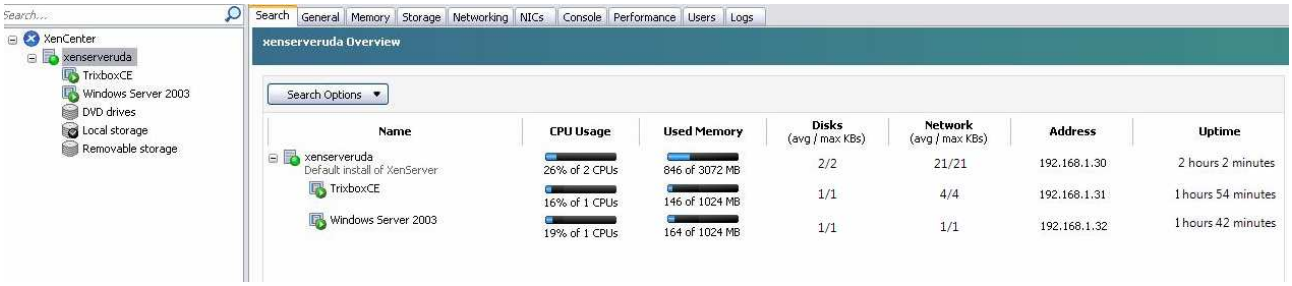

Figura 76: Estado del CPU, memoria, disco, red y tiempo de ejecución.

Además estos datos pueden ser accedidos a través del servidor XenServer por consola y observamos el uso de la CPU y de la memoria. Como se puede observar en la figura 77. Los cuales son similares a los presentados por el programa cliente.

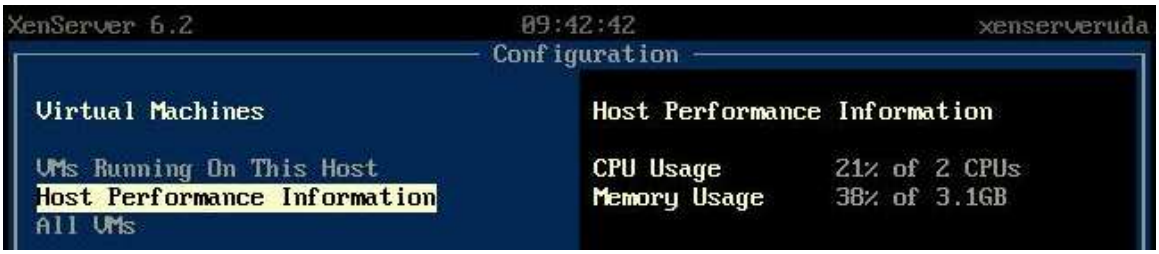

Figura 77: Uso del CPU y memoria desde la consola.

4.3.3 Rendimiento del CPU, Memoria y Red.

El software nos permite observar también el consumo de estos tres elementos mediante un gráfico de tiempo desde nuestro servidor. Como se puede observar en la figura 78, la mayor utilización del CPU y el incremento de memoria se dieron al momento de arrancar los sistemas operativos virtualizados, posteriormente el uso del CPU disminuyó notablemente y el uso de la

memoria se estabilizo. En cuanto a la red su uso se ve afectada por la relación a los pedidos de servicios a los servidores.

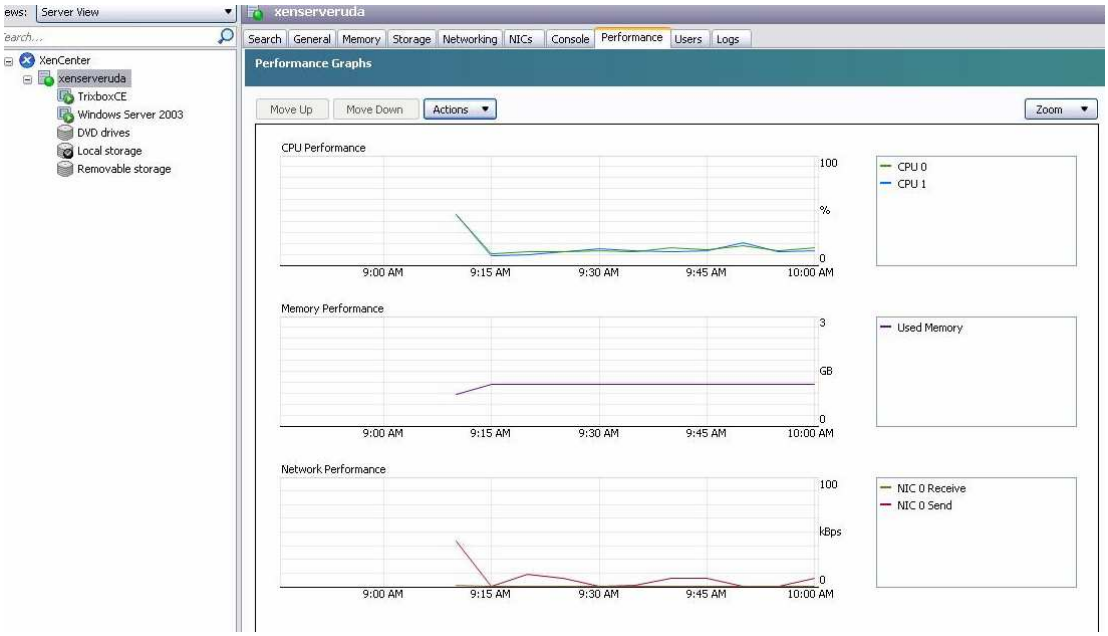

Figura 78: Rendimiento del CPU, memoria y la red.

Teniendo en cuenta todos los datos extraídos mediante los distintos tipos de software utilizados para realizar las pruebas de rendimiento y tests realizados a ambos métodos de virtualización, nos encontramos listos para generar conclusiones y recomendaciones válidas sobre el funcionamiento de estos sistemas y cual es el adecuado para el uso de una empresa.

Capítulo 5: Comparando los sistemas.

Estas comparaciones fueron tomadas de la propia experiencia al haber utilizado ambos sistemas de virtualización, completados con el conocimiento obtenido al implementar estas plataformas de confrontación relativa en este proyecto. Teniendo en cuenta los datos extraídos en los análisis realizados por el software EVEREST y mediante las estadísticas aportadas por el cliente XenCenter.

5.1 La Instalación.

Al momento de instalar sendos sistemas surge una clara diferencia, VMWare es un software que trabaja sobre un sistema anfitrión (en este caso sobre un sistema Windows XP), por lo que la instalación de ésta no causa ningún tipo de dificultad. Se hace de manera intuitiva, rápida y en pocos pasos.

En cuanto a XenServer, como vimos, es un sistema operativo que debe instalarse como tal en una PC destinada especialmente a este. Su instalación mucho más compleja, solicita diversos datos que se deben tener previamente estudiados ya que estos serán cruciales en el uso de nuestra red y de nuestros sistemas virtualizados. Además nuestro procesador debe de soportar la tecnología que requiere la paravirtualización, como ser un Intel con Intel VT o AMD con AMD-V. Estas APIs ofrecen instrucciones especiales que el software de virtualización puede emplear para permitir una ejecución más eficiente.

Una vez instalado el XenServer también tenemos que considerar que se requiere instalar en otra PC el software XenCenter con el cual controlaremos el servidor. No presenta dificultades su instalación pero su uso debe ser precavido a la hora de conectarse al servidor y realizar modificaciones desde este.

5.2 Creación de Máquinas Virtuales.

Cómo se demostró anteriormente con VMWare la creación de MV es sencilla y no presenta inconvenientes a la hora de decidir los recursos que se van a utilizar. En simples pasos como decidir

el tamaño del disco a utilizar, cantidad de núcleos del CPU, la memoria RAM y el tipo de conexión a Internet ya estamos listos para comenzar a usar la MV.

En cuanto a XenServer la creación de éstas, básicamente solicita los mismos datos que los anteriores, como diferencia más crítica podemos encontrar que al especificar el almacenamiento de la MV puede ser en el servidor o en otro dispositivo remoto que contenga un disco con mayor espacio disponible, por lo que estaríamos utilizando los recursos del servidor, el disco para almacenar la MV en otra Terminal y controlarla desde otro equipo con el XenCenter.

5.3 Funcionamiento de las MV.

Una vez instalados nuestros sistemas operativos virtualizados, con ambos métodos (virtualizados o paravirtualizados) la utilización de los mismos se presenta de eficientemente, se puede trabajar en cada uno de ellos sin apreciar que los equipos están virtualizados.

Al realizar las pruebas de funcionamiento tanto con Windows Server 2003, solicitando conexión al sistema de controlador de dominios, como con TrixBox conectando las llamadas IP, la reacción de los servidores fue perfecta sin observar detalles, para ninguno de los dos casos virtualizados o paravirtualizados.

El buen funcionamiento de los sistemas virtuales con ambos métodos se puede dar por los siguientes motivos a tener en cuenta, a cada sistema se les asigno 1 GB de memoria, lo cual es más que suficiente para el correcto funcionamiento de estos sistemas operativos (Windows Server 2003 y Trixbox). Además de no contar con una gran cantidad de equipos clientes que realicen peticiones a ambos servidores.

5.4 Rendimiento del equipo anfitrión.

Cómo pudimos observar de los datos tomados en los test realizados a ambos sistemas, lo más crítico es la utilización de memoria RAM, con el sistema de virtualización, con los dos servidores funcionando el uso de memoria supero el 50% de la misma. En cuanto a los sistemas paravirtualizados, en el mismo momento, el uso de memoria no había alcanzado el 40%. También

se debe tener en cuenta que el manejo del CPU es mejor realizado por el sistema de paravirtualización donde el uso aproximado de este es del 25%, mientras la virtualización ronda el 40%. Podemos observar estas comparaciones en la figura 79.

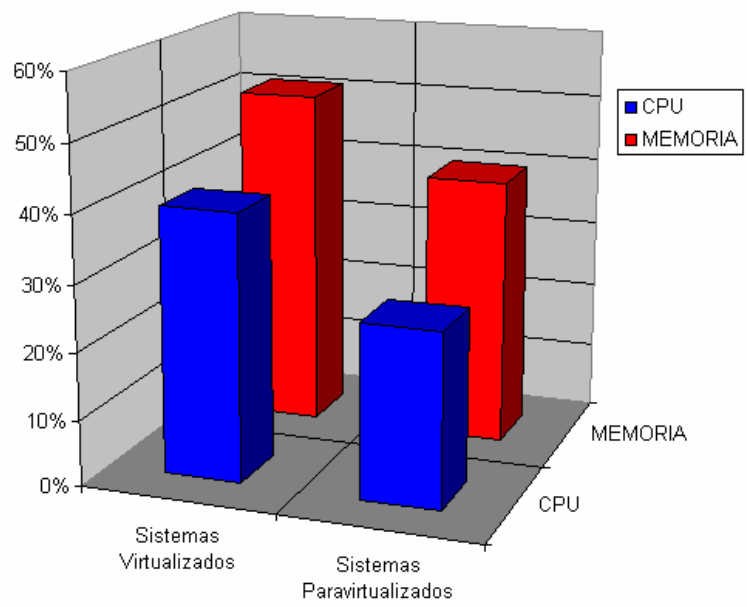

Figura 79: Consumo del CPU y Memoria con ambos métodos en el host anfitrión.

En cuanto al uso del CPU se observaba estable hasta el momento de realizar pruebas de llamadas a los servicios o durante las peticiones de los equipos terminales al controlador de dominio instalado en Windows Server 2003, con los sistemas virtualizados pudimos observar un mayor requerimiento del CPU que en los sistemas paravirtualizados.

Para realizar una comparación en el uso de la red deberíamos de tener una cantidad de usuarios mayor y estar realizando peticiones de red durante un gran lapso de tiempo para poder obtener un resultado significativo en su uso. Pero como solo se realizaron pruebas con dos equipos conectados al servidor la utilización de la red es casi imperceptible y no se podría tomar ningún dato para realizar una comparación de su uso en ambos sistemas virtuales.

5.5 Ventajas y desventajas del software utilizado.

Durante la aplicación de los métodos de virtualización, implementando el software ya mencionado surgen las siguientes ventajas y desventajas que estos proporcionan a la hora de su ejecución y administración:

Ventajas de VMware:

- Solidez, estabilidad, seguridad.
- Admite drivers en los entornos emulados.
- Esta indicado para consolidación de servidores e investigaciones técnicas.

Desventajas de VMware:

- Pobre rendimiento del gestor de maquinas virtuales con hardware de bajo rendimiento.
- La versión gratuita es de uso personal y no empresarial.
- Al actualizar el kernel se debe ejecutar nuevamente el instalador.

Ventajas de XenServer:

- Desempeño casi nativo.
- Permite virtualización completa y paravirtualización.
- Puede funcionar aun en hardware que no soporta virtualización completa.
- Permite la migración en caliente de los sistemas clientes.
- Existe la convivencia entre servidores virtualizados y servidores paravirtualizados.

Desventajas de XenServer:

- Los clientes deben ser modificados para su funcionamiento.
- No es compatible con la interfaz avanzada de configuración de energía (ACPI, APM) en tecnologías portátiles.
- No todo el hardware esta soportado.
- Problemas con algunos drivers propietarios.
- No admite varios chips de wlan y bridges cardbus.

Capítulo 6: Aporte Personal.

Antes de crear una infraestructura virtual se recomienda realizar una planificación adecuada con etapas de: evaluación, planificación, construcción y administración. Se debe realizar un estudio de consolidación de todos los servidores que van a formar parte de la infraestructura virtual, para tener un dimensionamiento apropiado de la capacidad necesaria que debe tener, considerando futuros crecimientos y recursos disponibles para cubrir necesidades de alta disponibilidad. Para ello cito los siguientes pasos a seguir al momento de decidir implementar un sistema de virtualización.

Primero: ¿Cuándo virtualizar?

Cuando una empresa necesita expandir sus horizontes tecnológicos y comenzar a brindar una mayor cantidad de servicios a sus clientes o empleados, necesita para ello contar con una mayor cantidad de servidores que proporcionen estos servicios.

Segundo: ¿Qué virtualizar?

El método más económico a la hora de afrontar esta situación es Virtualizar. Pero a la hora de hacer esto, debemos conocer las limitaciones que nuestra empresa posee. Todos los servicios pueden ser virtualizados o los de menores requerimientos.

Tercero: ¿Puedo utilizar mis recursos?

Si la empresa ya cuenta con un servidor lo suficientemente bueno como para poder virtualizar los sistemas que esta requiere brindar. El método de virtualización que ofrece VMWare Workstation es una alternativa sencilla y confiable. Teniendo en cuenta las limitaciones del hardware del equipo servidor, veremos cuantas maquinas virtuales podremos crear y si cumplen con lo solicitado.

Cuarto: ¿Puedo migrar mis sistemas a un sistema virtual?

Existen numerosas herramientas gratuitas y comerciales diseñadas para ayudarlo con la migración de sistemas entre los mundos físico y virtual (PlateSpin, PowerConvert, VMware Convert, Microsoft®Virtual Server Migration Toolkit y su correspondiente software de clonación); estas herramientas de migración, además, sirven para solucionar los potenciales problemas que surgen a partir de la no correspondencia de hardware entre el servidor físico y la máquina virtual.

Una de sus funciones es la de pasar los drivers necesarios al núcleo del sistema operativo para inicializar correctamente los drivers durante la fase de arranque del sistema.

Quinto: ¿Qué parámetros utilizar para elegir entre una tecnología y otra?

Si con la virtualización no se logran los objetivos fijados para el rendimiento y el correcto accionar de la empresa, y ésta, estaría dispuesta a invertir en un servidor con un hardware con mejores características, utilizar la paravirtualización presentaría un mejor desempeño del sistema y una mayor seguridad con las MV creadas.

Como aporte final puedo decir que la paravirtualización sería la mejor opción en la mayoría de los casos de una empresa en crecimiento, sino, si la empresa ya cuenta con un equipo servidor, que no soporte esta tecnología, se debería de probar si la virtualización cumplen con los parámetros solicitados y el funcionamiento de los servicios es el adecuado.

Capítulo 7: Conclusiones.

Cómo conclusiones al trabajo realizado podemos obtener que:

La virtualización es clave para el desarrollo de procesos en lo que se refiere al ahorro de espacio, energía, dinero y el poder utilizar los recursos necesarios sin dejar de lado la capacidad de la máquina. En esta nueva era, la virtualización se esta imponiendo con fuerza debido a que hoy en día muchos procesos dependen de la capacidad que tengan de hacer múltiples tareas, tomándolo como opción incluso para combatir la crisis económica actual.

El sistema de virtualización utilizado en este proyecto (VMware), presenta un uso sencillo y predecible a la hora de utilizar su entorno. Las MV se instalan sin mayor inconveniente, pero su funcionamiento afecta el rendimiento del host anfitrión de manera creciente mientras la utilización de estos servidores va en aumento.

En cuanto a la paravirtualización presenta como ventaja la mejora de rendimiento general de los dispositivos de entrada y salida, la CPU y la Memoria. Los sistemas operativos invitados y el anfitrión interactúan de manera más directa con los recursos físicos del computador.

Al estar instalado en un servidor dedicado, mejora en cuanto a la poca carga que le da al procesador al no tener que tener una capa completa de virtualización. Lo que introduce cambios en los sistemas operativos invitados permitiéndoles la comunicación directa con el hypervisor. Este concepto hace necesario que nuestro servidor tenga disponible un procesador con Intel VT o AMD-V.

Como conclusión final puedo decir que la paravirtualización supera las ventajas de la virtualización y mejora las desventajas que presenta.

Capítulo 8: Bibliografía

8.1 Libros

ROS, Josep; “Virtualización Corporativa con VMware”. Ncora Information Technology S.L. 2008.

TAKENURA Chris; CRAWFORD Luke S.. “The Book of Xen”. No Starch Press. 2010.

VMWARE. “Understanding Full Virualizatin, Paravirtualization, and Hardware Assist”. White Paper. 2007.

VON HAGEN William. “Professional Xen Virtualization”. Wiley Publishing, Inc. 2008.

WILLIAMS David E.; GARCIA Juan. “Virtualization with Xen”. SYNGRES. 2007.

8.2 Páginas Web

http://www.dte.eis.uva.es/Docencia/ETSII/SMP/BAK/Ha_SComp/historia.htm

http://www.multicians.org/thvv/360-67.html

http://wiki.xenproject.org/wiki/Paravirtualization_(PV)

http://networksandservers.blogspot.com.ar/2011/11/para-is-english-affix-of-greek-origin.html