

# **ESCUELA SUPERIOR POLITÉCNICA DEL LITORAL**

**Facultad de Ingeniería en Electricidad y Computación**

Informe de Materia de Graduación

## **“Implementación de un ambiente de Virtualización para el manejo de múltiples servidores de VoIP sobre una plataforma común de hardware”**

Previa a la obtención del Título de:

**LICENCIADO EN REDES Y SISTEMAS OPERATIVOS**

Presentada por:

**MARÍA BELÉN SOTAMINGA REYES**

**CARLOS LEOPOLDO GUERRERO VALAREZO**

**ALBERTO EDUARDO ABAD ERAS**

GUAYAQUIL – ECUADOR

AÑO 2011

# AGRADECIMIENTO

A Dios sobre todas las cosas, por guiar de la mejor manera nuestros pasos hacia este largo camino que hemos emprendido.

A nuestras amadas familias, por brindarnos todo el apoyo que hemos necesitado y ser imprescindibles cada momento de nuestra vida.

A nuestros profesores, amigos y a todas las personas que anónimamente aportan en Internet y comparten sus conocimientos y experiencias, con lo cual hemos conseguido formarnos profesionalmente.

# DEDICATORIA

*A Dios, a mis padres, a todos mis seres amados, pilares fundamentales de mi vida, porque por ellos trabajo por ser mejor cada día.*

**María B. Sotaminga Reyes**

*A Dios, a mis padres y a mis hermanos por todo el apoyo brindado en todo momento, a mi hijo Mauricio por animarme a seguir adelante y superarme siempre.*

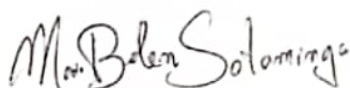
**Carlos. L. Guerrero Valarezo**

*A mi Dios, mi Esposa Evelin fuente de mi motivación, mi Madre por ser un ejemplo para mi Vida, mi Padre, mi Abuelita y demás familiares por sus palabras y cuidados.*

**Alberto E. Abad Eras**

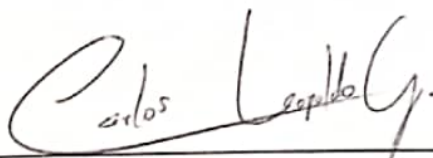
# DECLARACIÓN EXPRESA

"La responsabilidad del contenido de esta Trabajo de Grado, nos corresponde exclusivamente; y el patrimonio intelectual de la misma a la Escuela Superior Politécnica del Litoral". (Reglamento de Graduación de la ESPOL)



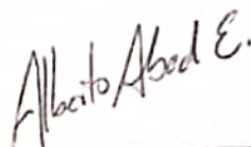
---

María Belén Sotaminga Reyes



---

Carlos Leopoldo Guerrero Valarezo



---

Alberto Eduardo Abad Eras

# TRIBUNAL DE SUSTENTACIÓN



---

Ing. Gabriel Astudillo

PROFESOR DE LA MATERIA DE GRADUACIÓN



---

Ing. Gonzalo Luzardo

PROFESOR DELEGADO POR EL DECANO DE LA FACULTAD

# RESUMEN

El proyecto desarrollado consistió en implementar un ambiente virtualizado permitiendo el manejo de varios servidores de VoIP sobre una misma plataforma de hardware, valiéndonos de CentOS 5.5 como sistema operativo residente y de su módulo KVM (Kernel-based Virtual Machine) como solución de virtualización.

KVM permite ejecutar máquinas virtuales utilizando imágenes de disco que contienen sistemas operativos sin modificar. Cada máquina virtual tiene su propio hardware virtualizado: una tarjeta de red, discos duros, tarjetas gráfica, etc.

Se configuró también una centralita telefónica basada en el software libre Asterisk para el Servidor físico y Elastix para las máquinas virtuales.

Lo que se presenta en este documento son todos los pasos necesarios para montar un Servidor de maquinas virtuales por medio de KVM y las herramientas necesarias para poder administrar tanto al servidor como así también a sus respectivas maquinas virtuales.

## INDICE DE CONTENIDO

### INTRODUCCIÓN

CAPITULO:1 ANTECEDENTES Y JUSTIFICACIÓN .....	1
1.1. ANTECEDENTES .....	2
1.2. JUSTIFICACIÓN.....	3
1.3. DESCRIPCIÓN DEL PROYECTO.....	3
1.3.1. Objetivos Generales.....	4
1.3.2. Objetivos Específicos.....	4
1.4. METODOLOGIA .....	5
1.5. PERFIL DE LA TESIS.....	5
CAPITULO II:VIRTUALIZACIÓN, KVM Y ELASTIX .....	7
2.1. HISTORIA DE LA VIRTUALIZACIÓN .....	8
2.2. DEFINICIÓN DE VIRTUALIZACIÓN .....	10
2.3. FACTORES A CONSIDERAR PARA LA VIRTUALIZACIÓN .....	12
2.4. VENTAJAS Y DESVENTAJAS DE LA VIRTUALIZACIÓN .....	12
2.4.1. Ventajas .....	13
2.4.2. Desventajas .....	14
2.5. TIPOS DE VIRTUALIZACIÓN.....	15
2.5.1. Virtualización a nivel del Sistema Operativo .....	15
2.5.2. Paravirtualización .....	16
2.5.3. Virtualización completa.....	17
2.5. KERNEL-BASED VIRTUAL MACHINES - KVM .....	18
2.5.1. Por qué escoger KVM.....	19

2.5.2.	Características .....	19
2.6.	ELASTIX .....	21
2.6.1.	Comunicaciones Unificadas .....	21
2.6.2.	Características y Funcionalidades de Elastix .....	22
2.6.2.1.	PBX.....	22
2.6.2.2.	FAX.....	24
2.6.2.3.	General.....	25
2.6.2.4.	EMAIL .....	25
CAPÍTULO 3:IMPLEMENTACIÓN .....		26
3.1.	INTRODUCCIÓN .....	27
3.2.	HARDWARE.....	27
3.2.1.	Servidor .....	28
3.2.1.	Teléfono IP .....	29
3.3.	SOFTWARE .....	30
3.3.1.	Servidor PBX.....	30
3.3.2.	Máquinas Virtuales .....	31
3.4.	CONFIGURACIÓN DE ARCHIVOS DE ASTERISK .....	32
3.4.1.	Configuración de ETC/DAHDI/SYSTEMS.CONF .....	32
3.4.2.	Configuración DAHDI-CHANNELS.CONF .....	33
3.4.3.	Configuración de CHAN_DAHDI.CONF .....	34
3.4.4.	Configuración de SIP.CONF .....	35
3.4.5.	Configuración de EXTENSIONS.CONF .....	38
3.4.	INSTALACIÓN DE KVM.....	41
3.4.1.	Preparación .....	41



3.4.2.	Instalando KVM.....	42
3.4.3.	Configuración Bridge para usar KVM.....	43
CAPITULO IV:FUNCIONAMIENTO Y PRUEBAS .....		47
4.1.	INICIANDO VIRTUAL MACHINE MANAGER .....	48
4.2.	CREANDO UNA MÁQUINA VIRTUAL.....	49
4.2.1.	Configurando el Método de Instalación .....	50
4.2.2.	Configurando el almacenamiento virtual KVM.....	51
4.2.3.	Configurando las opciones de redes KVM .....	52
4.2.4.	Configuración de las opciones de memoria de la máquina virtual KVM y la CPU.....	54
4.3.	INICIALIZANDO LA NUEVA MÁQUINA VIRTUAL .....	55
4.4.	CONFIGURANDO ELASTIX .....	57
4.4.1.	Configuración del Trunk SIP .....	58
4.4.2.	Configuración del Trunk IAX2.....	60
4.5.	CONFIGURACIÓN DEL TELÉFONO IP CISCO SPA 303.....	63
4.6.	PRUEBAS.....	64
CONCLUSIONES .....		67
RECOMENDACIONES.....		68
GLOSARIO DE TERMINOS .....		70
ANEXOS .....		72
RENDIMIENTO DURANTE EL PROCESO DE INSTALACIÓN DE ELASTIX EN LAS VM		
BILIOGRAFIA .....		76

## ÍNDICE DE FIGURAS

Fig. 2.1 Gestión de hardware .....	12
Fig. 2.2 Ventajas de Virtualizar .....	13
Fig. 2.3 Virtualización del SO .....	16
Fig. 2.4 Paravirtualización .....	16
Fig. 2.5 Full Virtualization .....	17
Fig. 2.6 Características KVM .....	20
Fig. 2.7 Funcionalidades de Elastix .....	22
Fig. 4.1 Virtual Machine Manager .....	48
Fig. 4.2 Creando un nuevo sistema virtual .....	49
Fig. 4.3 Creando un nuevo sistema virtual .....	50
Fig. 4.4 Almacenamiento en KVM .....	52
Fig. 4.5 Red en KVM .....	53
Fig. 4.6 Memoria y CPU en KVM .....	55
Fig. 4.7 Instalando Elastix en la nueva VM .....	56
Fig. 4.8 Salida del SIP SHOW PEERS .....	60
Fig. 5.1 Testing con SIPP .....	65
Fig. 5.2 Uso de Memoria y CPU .....	66
Fig.3.1 Cisco IPPHONE 303 .....	29

## ÍNDICE DE TABLAS

Tabla I Características del Servidor .....	29
Tabla II Software del Servidor PBX .....	30
Tabla III Características de las Máquinas Virtuales .....	31
Tabla IV Hardware asignado a las VM .....	32

# INTRODUCCIÓN

En los últimos años, se ha acelerado el fenómeno de virtualización en la docencia universitaria como un reflejo de la presencia extensiva e intensiva de las Nuevas Tecnologías de Información y Comunicación que vienen generando cambios paradigmáticos en la educación superior. Las universidades de los países en desarrollo enfrentan el desafío de servir a una población cada vez mayor de estudiantes. El uso de la tecnología de virtualización (parcial o total) puede ser un factor transformador de sus estructuras y funciones, un instrumento para mejorar su cobertura, calidad, pertinencia y equidad de acceso como una manera de construir una nueva identidad en la Sociedad del Conocimiento.

Actualmente la virtualización está asociada a conceptos como optimización, seguridad, escalabilidad y facilidad de la gestión. Los avances en procesadores han llevado a una situación en la que sólo se aprovecha el 20-30% de su potencia<sup>1</sup>, la idea principal es la de permitir ejecutar varios sistemas operativos simultáneamente sobre el mismo hardware. Para ello separa las dos funciones básicas que realiza un sistema de tiempo compartido: multiprogramación y abstracción del hardware donde la virtualización se convierte en una seria alternativa para alcanzar un objetivo inminente: reducir el coste total de operación (TCO) de la plataforma de servidores.

---

<sup>1</sup> Según Gartner Group (<http://www.gartner.com/technology>)

# **CAPITULO I**

## **ANTECEDENTES Y JUSTIFICACIÓN**

## 1.1. ANTECEDENTES

Los estudiantes para el desarrollo de sus actividades y fortalecimiento de habilidades y destrezas en las plataformas cliente/servidor, demandan contar con prácticas en las que se pueda adquirir una experiencia en “entorno real”, solicitando servicios a equipos reales sin afectar el buen funcionamiento de los mismos. Diversos proyectos de investigación elaborados por los usuarios, son generalmente desarrollados en un mismo servidor físico instalado y configurado por lo que generalmente son desconfigurados produciendo la suspensión de sus servicios, pérdida de datos y empleando horas adicionales para reinstalar y reconfigurar los mismos.

Una solución para estos inconvenientes, sería la adquisición de varios servidores, la cual implica una alta inversión económica por el costo de la infraestructura física, consumo de electricidad, así como la capacitación que debe recibir el personal encargado del mantenimiento de los mismos, siendo estos gastos sólo una parte pequeña del coste total. Y es por esta razón que la virtualización y optimización de recursos es tan importante. La virtualización “esconde” las características físicas de un ordenador a los usuarios, aplicaciones o ambos.

## **1.2. JUSTIFICACIÓN**

Para contribuir en el proceso de enseñanza-aprendizaje se necesita trabajar con herramientas de apoyo académico en plataformas de distribución libre, es decir de uso exclusivamente pedagógico, más no de producción, promoviendo en los usuarios el uso de plataformas bajo Licencia Pública General (General Public License-GPL) porque rindan mucha flexibilidad y proyección a futuro.

La Virtualización es sin duda uno de los temas más importantes a nivel tecnológico para realizar pruebas de laboratorios y hasta gestionar directamente servidores sin tener que contar con mayor cantidad de servidores físicos en una red.

## **1.3. DESCRIPCIÓN DEL PROYECTO**

La implementación del presente proyecto pretende alcanzar los siguientes objetivos:

### 1.3.1. Objetivos Generales

- Implementar un sistema que nos permita contener múltiples instancias de servidores de Voz sobre IP (VoIP) en una misma plataforma de hardware como un Entorno Virtual (EV), logrando así la optimización del mismo.
- Determinar un método para dividir recursos computacionales en múltiples ambientes independientes.

### 1.3.2. Objetivos Específicos

- Implementar un sistema virtual para el laboratorio de simulación de Telecomunicaciones, para optimizar el uso del hardware VOIP.
- Dotar a cada máquina virtual de acceso a los recursos del hardware, usuario, dirección IP, procesos, archivos, aplicaciones, sistema de librerías, archivos de configuración, etc.

El proyecto a realizar consiste en la implementación de una solución de virtualización basada en *Kernel-Based Virtual Machines* (KVM) que

nos permitirá ahorrar costes de hardware (HW), alojamiento de equipos y aprovechamiento de recursos.

#### **1.4. METODOLOGIA**

Para cumplir nuestros objetivos instalaremos nuestras máquinas virtuales en KVM sobre un servidor con el sistema operativo CentOS. Instalaremos en cada máquina virtual (VM) una centralita telefónica basada en ELASTIX para el flujo de llamadas salientes desde la red de telefonía IP.

Dotaremos a las VM de conexión con la tarjeta análoga Digium a través de ASTERISK que será previamente instalado en el equipo host, mientras que las llamadas entre las máquinas virtuales (VM) serán establecidas por los protocolos SIP e IAX.

#### **1.5. PERFIL DE LA TESIS**

Nuestro proyecto tiene como objetivo principal implementar un sistema virtualizado optimizando el hardware del laboratorio de Telecomunicaciones, para permitir a los estudiantes acceder a un equipo virtual con una central VoIP y analizar en entorno real todos los beneficios que ofrece.



En el capítulo 2, detallamos las bases teóricas sobre virtualización como sus tipos, características, aplicaciones, tipos de implementación, y beneficios que proporciona actualmente.

En el capítulo 3 se indican las especificaciones técnicas de la solución, el análisis, y el procedimiento de implementación del proyecto.

Por último, en el capítulo 4, realizamos las pruebas de conexión, el establecimiento de las llamadas, evaluación del performance del hardware y escalabilidad del mismo.

# **CAPITULO II:**

**VIRTUALIZACIÓN, KVM Y ELASTIX**

## 2.1. HISTORIA DE LA VIRTUALIZACIÓN

Fue IBM quien empezó a implementar la virtualización hace más de 30 años como una manera lógica de particionar ordenadores mainframe en máquinas virtuales independientes. Estas particiones permitían a los mainframes ejecutar varias aplicaciones y procesos al mismo tiempo.

La virtualización se abandonó de hecho en las décadas de los ochenta y los noventa, cuando las aplicaciones cliente-servidor y los servidores x86 y escritorios económicos establecieron el modelo de informática distribuida. La amplia adopción de Windows y la emergencia de Linux como sistemas operativos de servidor en los años noventa convirtieron a los servidores x86 en el estándar del sector. El incremento de implementaciones de servidores x86 y escritorios generó nuevos problemas operacionales y de infraestructura de TI. Entre estos problemas se incluyen los siguientes:

Baja utilización de la infraestructura. Las implementaciones típicas de servidores x86 logran un promedio de utilización de entre un 10 y un 15% de la capacidad total, según señala International Data Corporation (IDC)<sup>2</sup>,

---

<sup>2</sup> International Data Corporation IDC – Introducción a la Virtualización  
<http://www.slideshare.net/frikinika/idc-introduccionmercadovirtualizacion>

una empresa de estudios de mercado. Normalmente, las organizaciones ejecutan una aplicación por servidor para evitar el riesgo de que las vulnerabilidades de una aplicación afecten a la disponibilidad de otra aplicación en el mismo servidor.

Incremento de los costes de infraestructura física. Los costes operativos para dar soporte al crecimiento de infraestructuras físicas han aumentado a ritmo constante. La mayor parte de las infraestructuras de TI deben permanecer operativas en todo momento, lo que genera gastos en consumo energético, refrigeración e instalaciones que no varían con los niveles de utilización.

Incremento de los costes de gestión de TI. A medida que los entornos se hacen más complejos, aumenta el nivel de especialización de la formación y la experiencia que necesita el personal de gestión de infraestructuras y, consecuentemente, aumentan los costes asociados a dicho personal. Las organizaciones gastan cantidades desproporcionadas de dinero y recursos en tareas manuales ligadas al mantenimiento de los servidores, y aumenta la necesidad de personal para realizarlas.

Insuficiente protección ante desastres y fallas. Las organizaciones se ven cada vez más afectadas por las interrupciones del servicio de las aplicaciones de servidor críticas y la falta de acceso a escritorios de usuario final fundamentales. La amenaza de ataques a la seguridad, desastres

naturales, pandemias y terrorismo han acentuado la importancia de la planificación de la continuidad del negocio, tanto en lo relativo a escritorios como a servidores.<sup>3</sup>

Escritorios de usuario final de mantenimiento elevado. La gestión y la seguridad de los escritorios empresariales plantean numerosos desafíos. Controlar un entorno de escritorios distribuidos y aplicar políticas de gestión, acceso y seguridad sin perjudicar la capacidad del usuario de trabajar con eficacia es complejo y costoso. Se tienen que aplicar continuamente muchos parches y actualizaciones en el entorno del escritorio para eliminar los riesgos de seguridad

## **2.2. DEFINICIÓN DE VIRTUALIZACIÓN**

La tecnología de virtualización tuvo sus inicios en la década de los sesenta para particionar mainframes de gran tamaño a fin de mejorar su utilización. Actualmente, los ordenadores basados en la arquitectura x86 enfrentan los mismos problemas de rigidez e infrautilización enfrentados por los mainframes en aquella década.

---

<sup>3</sup> Muy Pymes, "Virtualización", <http://www.muypymes.com/tecnologia/software/4605-como-funciona-la-virtualizacion-y-para-que-sirve.html>

Existen algunos tipos de herramientas de virtualización, pero todos tienen algo en común, se trata de herramientas que pueden hacer creer a otros programas que son el hardware y software que necesitan.

El objetivo de la virtualización es tener uno a varios sistemas operativos sobre uno ya existente, permaneciendo este sin verse afectado y pudiendo arrancarlos de manera independiente a diferencia de la instalación en el mismo equipo gracias a una capa de software llamada *Virtual Machine Monitor* o VMM que crea una capa de abstracción entre el equipo físico o *host* y el software del sistema operativo de la máquina virtual o *guest*.

Esta capa gestiona el hardware los cuatro elementos más importantes de la computadora: CPU, memoria, red y almacenamiento como podemos observar en la figura 2.1



Fig. 2.1 Gestión de hardware

### 2.3. FACTORES A CONSIDERAR PARA LA VIRTUALIZACIÓN

En las tecnologías de virtualización se consideran como factores: reducción de costes, mejora el retorno de las inversiones de las TI casi inmediato, uso racional del hardware mayor flexibilidad, reducción de gastos operativos, reducción en el consumo de energía, mayor eficiencia de los recursos informáticos, una gestión y administración de los recursos más ágil y centralizada, aumenta la capacidad de los servidores entre un 16 y un 80 por ciento dependiendo de las características técnicas de hardware.

### 2.4. VENTAJAS Y DESVENTAJAS DE LA VIRTUALIZACIÓN

### 2.4.1. Ventajas

Son innumerables las ventajas que ofrece la virtualización, pero a continuación puntualizamos las más importantes:

- Ahorra reinicio en caso de que tengamos que cambiar habitualmente de S.O.
- Facilita la incorporación de nuevos recursos de hardware para los servidores virtualizados.



Fig. 2.2 Ventajas de Virtualizar

- Reduce costes de mantenimiento y consumo eléctrico.
- Reduce los costes de IT gracias al aprovechamiento de recursos.
- Administración centralizada que simplifica la gestión de procesamiento, memoria, red y almacenamiento disponible en nuestra infraestructura



- Aislamiento : un fallo general de sistema de una máquina virtual no afecta al resto de máquinas virtuales
- Proporcionando un consumo de recursos homogéneo y óptimo en toda la infraestructura

La figura 2.2 describe las características descritas en este apartado.

### **2.4.2. Desventajas**

Hay ciertos factores que debemos tener en consideración que podrían ser vistos negativamente:

- Un sistema virtualizado nunca alcanzará el mismo rendimiento comparado con el que estuviera directamente instalado, porque que el hypervisor introduce una capa intermedia en la gestión del hardware por lo que el rendimiento de la máquina virtual se ve afectado irremediablemente.
- Si se daña el disco duro, se perderán todas las máquinas virtuales. Sugerimos uso del RAID, los discos no se dañan siempre, pero a veces pasa.

- Si nos roban la máquina, nos roban todas las máquinas virtuales. Sugerimos realizar respaldos.
- En fin, cualquier evento que ocurra con el hardware, afectará a todas las máquinas virtuales (corriente, red, etc) así que necesitamos un sistema bien redundante (doble red, doble disco, doble fuente de corriente, etc).

## **2.5. TIPOS DE VIRTUALIZACIÓN**

Tenemos los siguientes tipos de virtualización:

### **2.5.1. Virtualización a nivel del Sistema Operativo**

En este esquema no se virtualiza el hardware y se ejecuta una única instancia del sistema operativo (*kernel*). Los distintos procesos pertenecientes a cada servidor virtual se ejecutan aislados del resto, como lo muestra la figura 2.3 a continuación:

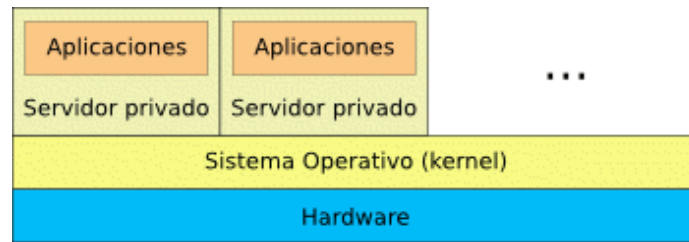


Fig. 2.3 Virtualización del SO

Su ventaja radica en la separación de los procesos de usuario prácticamente sin pérdida en el rendimiento, pero al compartir todos los servidores virtuales el mismo kernel no pueden obtenerse el resto de las ventajas de la virtualización.

### 2.5.2. Paravirtualización

Como se ilustra en la figura 2.4, la paravirtualización consiste en ejecutar sistemas operativos *guests* sobre otro sistema operativo que actúa como *hypervisor (host)*. Los *guests* tienen que comunicarse con el *hypervisor* para lograr la virtualización.

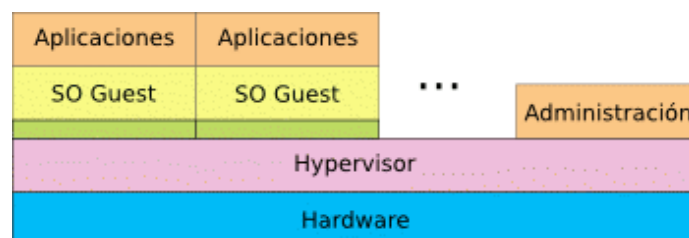


Fig. 2.4 Paravirtualización

Las ventajas de este enfoque son un muy buen rendimiento y la posibilidad de ejecutar distintos sistemas operativos como *guests*. Se obtienen, además, todas las ventajas de la virtualización enunciadas anteriormente. Su desventaja es que los sistemas operativos *guests* deben ser modificados para funcionar en este esquema.

### 2.5.3. Virtualización completa

La virtualización completa es similar a la paravirtualización pero no requiere que los sistemas operativos *guest* colaboren con el *hypervisor*. En plataformas como la x86 existen algunos inconvenientes para lograr la virtualización completa, que son solucionados con las últimas tecnologías propuestas por AMD e Intel.

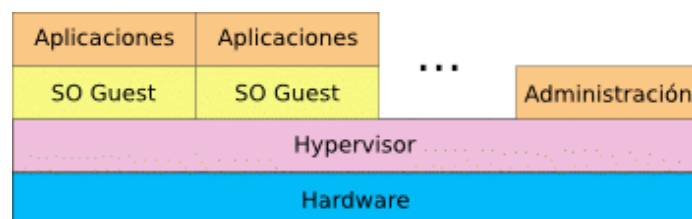


Fig. 2.5 Full Virtualization

Este método tiene todas las ventajas de la paravirtualización, con el añadido de que no es necesaria ninguna modificación a los *guests*. La única restricción es que estos últimos deben soportar la arquitectura de hardware utilizada. En la figura 2.5 podemos observar lo descrito.

## 2.5. KERNEL-BASED VIRTUAL MACHINES - KVM

KVM (Kernel-Based Virtual Machines), es una solución de virtualización completa sobre GNU/Linux que se incluyó oficialmente en el kernel 2.6.20 y consiste en un módulo para el kernel "kvm.ko" que proporciona al núcleo la infraestructura de virtualización y tiene un módulo por cada tipo de procesador, sea este INTEL o AMD.

- INTEL "kvm-intel.ko"
- AMD "kvm-amd.ko"

Se completa con la versión de QEMU que proporciona toda funcionalidad de Hardware Virtual para los sistemas virtualizados. Todas las nuevas tecnologías de desarrollo se están llevando a cabo en KVM, ya que va hacer la solución más recomendada y soportada por varios sistemas operativos como GNU/Linux, BDS, Solaris, MAC y Windows. También tiene mayor seguridad entre las maquinas virtuales y una configuración más amigable para los administradores.

### **2.5.1. Por qué escoger KVM**

A la hora de decidirnos por KVM, encontramos respuestas técnicas, económicas, y de todo tipo para responder a esta pregunta. Algunas técnicas son por ejemplo que KVM es un módulo más del kernel. Esto significa que no es un kernel diferente como pasa con Xen. Al ser un módulo del kernel podemos cargarlo y descargarlo en tiempo de ejecución y soporta otros módulos. Además, debido a que es parte del kernel, “hereda” muchas de las ventajas que ya tiene el kernel por sí como escalabilidad, soporte para diferentes procesadores (Itanium, x86- 64 y x86) y gestión de memoria Non-Uniform Memory Architecture (NUMA).

A continuación detallamos más características que resultan ventajosas sobre KVM.

### **2.5.2. Características**

- Está diseñado para procesadores x86, centrandose en la virtualización total.
- Kernel de GNU/Linux no modificado.
- Módulo que no necesita parches.

- Soporte de paravirtualización.
- KVM funciona en todo tipo de máquinas, servidores, escritorio o laptop.
- Permite la migración en caliente de máquinas virtuales. (fig. 2.7)



**Fig. 2.6 Características KVM**

- Administración vía web, grafica y consola por lo que resulta más amigable.
- Podemos utilizar el comando kill para matar procesos de las máquinas virtuales.
- Configuraciones de red tipo: Bridge, Route, NAT.
- Independencia al ejecutar múltiples maquinas virtuales ya que cada una tiene su propia instancia.

## **2.6. ELASTIX**

Elastix es un software de código abierto para el establecimiento de comunicaciones unificadas. Implementa gran parte de su funcionalidad sobre 4 programas de software muy importantes como son Asterisk, Hylafax, Openfire y Postfix.

### **2.6.1. Comunicaciones Unificadas**

Debido a que la telefonía es el medio tradicional que ha liderado las comunicaciones durante el siglo pasado, muchas empresas y usuarios centralizan sus requerimientos únicamente en sus necesidades de establecer telefonía en su organización confundiendo distros de comunicaciones unificadas con equipos destinados a ser centrales telefónicas. Sin embargo Elastix no solamente provee telefonía, integra otros medios de comunicación para hacer más eficiente y productivo su entorno de trabajo. Elastix incluye en su solución los siguientes medios de comunicación ilustrados en la figura 2.7.



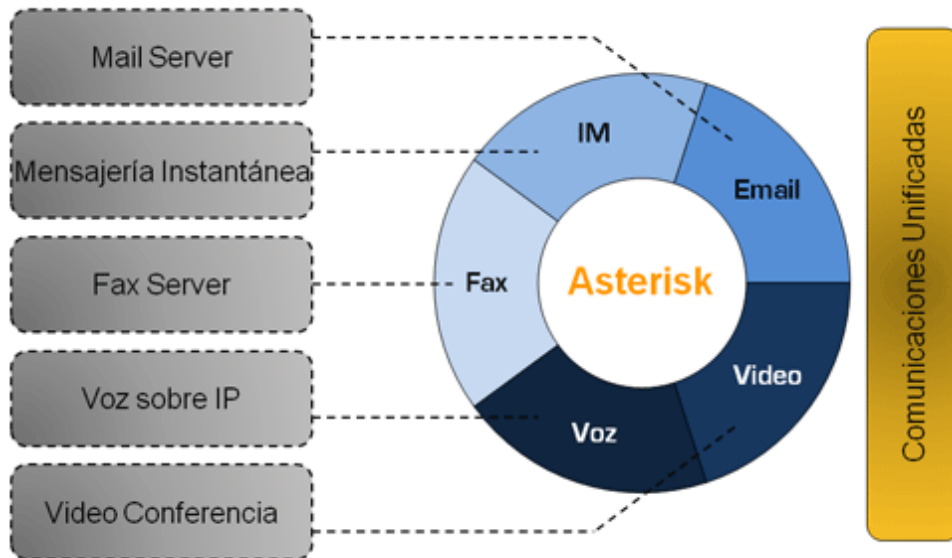


Fig. 2.7 Funcionalidades de Elastix

## 2.6.2. Características y Funcionalidades de Elastix

Elastix tiene múltiples características y funcionalidades relacionadas con los servicios que presta: Telefonía IP, Servidor de Correo, Servidor de Fax, Conferencias, Servidor de Mensajería Instantánea, entre otros. Nuevas características, funcionalidades y servicios son añadidos en el desarrollo de nuevas versiones.

A continuación detallamos sus características y funcionalidades:

### 2.6.2.1. PBX

- Grabación de Llamadas
- Correo de Voz

- Correo de voz-a-Email
- IVR Configurable y Flexible
- Soporte para Sintetización de Voz
- Herramienta para la creación de extensiones por lote
- Cancelador de eco integrado
- Provisionador de Teléfonos vía Web
- Soporte para videofonos
- Interfaz de detección de Hardware
- Servidor DHCP para asignación dinámica de Ips
- Reporte de detalle de llamadas (CDR)
- Tarifación con reporte de consumo por destino
- Reportes de uso de canales
- Soporte para colas de llamadas
- Centro de Conferencias con Salas Virtuales
- Soporte para protocolos SIP e IAX, entre otros
- Codecs soportados: ADPCM, G.711 (A-Law &  $\mu$ -Law), G.722, G.723.1 (pass through), G.726, G.729, GSM, iLBC (opcional) entre otros.
- Soporte para Interfaces Análogas como FXS/FXO (PSTN/POTS)
- Soporte para interfaces digitales E1/T1/J1 a través de los protocolos PRI/BRI/R2
- Identificación de llamadas (Caller ID)

- Troncalización
- Rutas entrantes y salientes con configuración por coincidencia de patrones de marcado
- Soporte para follow-me
- Soporte para grupos de timbrado
- Soporte para paging e intercom
- Soporte para condiciones de tiempo
- Soporte para PINes de seguridad
- Soporte para DISA (Direct Inward System Access)
- Soporte para Callback
- Soporte para interfaces tipo bluetooth a través de teléfonos celulares (chan\_mobile)

#### **2.6.2.2. FAX**

- Servidor Fax basado en HylaFax
- Personalización de faxes-a-email
- Visor de faxes integrado con PDFs descargables
- Control de acceso para clientes de fax
- Aplicación fax-a-email
- Puede ser integrada con Winprint Hylafax

### **2.6.2.3. General**

Ayuda en línea embebida

- Administración Centralizada de Actualizaciones
- Monitor de Recursos del Sistema
- Soporte para backup/restore a través de Web
- Configurador de parámetros de red
- Soporte para temas o skins
- Control de apagado/re-encendido de la central via Web
- Soporte para configuración de fechas en el servidor, horas y zonas horarias
- Control de Acceso a la Interfaz, basado en ACLs

### **2.6.2.4. EMAIL**

- Servidor de Email con soporte multidominio
- Soporte para cuotas
- Administración centralizada vía Web
- Soporte Antispam
- Interfaz de configuración de Relay
- Basado en Postfix para un alto volumen de correos
- Cliente de Email basado en Web

# **CAPÍTULO 3**

## **IMPLEMENTACIÓN**

### **3.1. INTRODUCCIÓN**

Esta solución está basada en el módulo de kernel KVM, que permite implementar varias instancias de sistemas operativos, reduciendo los costos que conllevan la instalación y mantenimiento de equipos físicos, además KVM por ser de libre acceso lo utilizamos en forma gratuita, cumpliendo con el objetivo de dar a conocer las bondades del software libre.

Instalaremos en cada una de las máquinas virtuales una PBX con Elastix, que al igual que KVM es software libre a fin de establecer las comunicaciones entre ellas. Elastix también incluye muchas funcionalidades que nos permiten ampliar sus usos para mejor aprovechamiento del hardware.

El fin del proyecto de graduación es aplicar los beneficios del software libre, ya que nos permite disminuir costos al implementar soluciones innovadoras en nuestro entorno estudiantil o laboral.

### **3.2. HARDWARE**

Las características básicas que debe cumplir el equipo servidor para el correcto funcionamiento de esta solución son:

- Mínimo 1GB de memoria RAM (recomendado 2 GB).
- Disco duro 320 GB.
- Procesador Intel o AMD con soporte para virtualización.
- Particiones independientes:

*/boot*

*swap*

*/tmp*

*/home*

*/*

*/var*

- Partición independiente para las máquinas virtuales:

*/vm*

Por otra parte tampoco podemos escatimar esfuerzos a la hora de escoger el hardware porque mucho del funcionamiento y rendimiento del host y de los guests dependen del equipo físico en que es montado.

### **3.2.1. Servidor**

Los requisitos de hardware del servidor que desarrollará nuestro proyecto son los que detallamos a continuación:

**Tabla I Características del Servidor**

Procesador	Intel Core2Duo E6750 2.66GHz
Mainboard	Intel DG33BU
RAM	2 Gb DDR2
Disco Duro	160 Gb
Tarjeta de Red	Ethernet Gigabit Intel 82566DC
Tarjeta análoga	Digium TDM400 2FXO 2FXS

### 3.2.1. Teléfono IP

#### CISCO IPPHONE 303



**Fig.3.1 Cisco IPPHONE 303**

El teléfono IP de 3 líneas Cisco SPA 303 (fig.3.1) con interruptor de 2 puertos, basado en el protocolo SIP ha sido probado para garantizar la



interoperabilidad integral con los equipos de los líderes en infraestructura de VoIP.

### 3.3. SOFTWARE

#### 3.3.1. Servidor PBX

El equipo que será utilizado como servidor de virtualización contendrá una centralita telefónica con los siguientes componentes instalados:

Tabla II Software del Servidor PBX

Sistema Operativo	Linux CentOS 5.5
Software IP PBX	Asterisk versión 1.8
Protocolos Configurados	SIP, DAHDI
KVM	kvm-83-164.el5_5.25

Para que KVM pueda funcionar correctamente, debemos instalar los siguientes paquetes:

- kmod-kvm: módulo(s) kvm para el kernel
- kvm: Kernel-based Virtual Machine

- `kvm-qemu-img`: Utilidad de imágenes de disco Qemu
- `kvm-tools`: Herramientas para debugging de KVM y de diagnóstico
- `python-virtinst`: Módulos de Python y utilidades para instalar máquinas virtuales
- `virt-manager`: Virtual Machine Manager (aplicación GUI, para instalar y configurar máquinas virtuales)
- `virt-viewer`: Virtual Machine Viewer (otra aplicación ligera para ver la consola de la máquina virtual o instalar Máquinas Virtuales)
- `bridge-utils`: Utilidades para configurar el Linux Ethernet bridge (se recomienda para las redes de KVM)

### 3.3.2. Máquinas Virtuales

En la virtualización utilizamos el modo Full Virtualizado, para poder instalar los sistemas operativos sin cambios en su kernel. Las máquinas virtuales que correremos tendrán las siguientes características de software instaladas.

**Tabla III Características de las Máquinas Virtuales**

Software IP PBX	Elastix 2.0.3
Protocolos configurados	SIP, IAX

## Hardware virtualizado

Tabla IV Hardware asignado a las VM

Memoria RAM	200 MB
Disco Duro	30 GB
Red	Ethernet modo Bridge
Sonido	No

### 3.4. CONFIGURACIÓN DE ARCHIVOS DE ASTERISK

#### 3.4.1. Configuración de ETC/DAHDI/SYSTEMS.CONF

El parámetro *fxsks=3* se compone de 3 partes, la primera *fxs* (*Foreign eXchange Station*) indica que ese canal está destinado para usarse con un teléfono analógico (FXS), el parámetro *ks* que significa *keystart signalling* - determina si un canal está abierto o cerrado, posee una mayor inteligencia y es más eficiente a la hora de detectar una desconexión.

La ultima parte =3 identifica cual es el canal que se utilizara en el dialplan (extensions.conf), en este ejemplo el canal 3 hace referencia al canal DAHDI/3 el cual se usara en /etc/asterisk/extensions.conf.

Configuración final:

```
fxoks=1
echocanceller=mg2,1
fxoks=2
echocanceller=mg2,2
fxsks=3
echocanceller=mg2,3
fxsks=4
echocanceller=mg2,4
```

### 3.4.2. Configuración DAHDI-CHANNELS.CONF

Configuración final dahdi-channels.conf:

```
;;; line="1    WCTDM/4/0
FXOKS"
signalling=fxo_ks
callerid="Channel 1" <4001>
mailbox=4001
group=5
context=from-internal
channel => 1
callerid=
mailbox=
group=
context=default

;;; line="2    WCTDM/4/1
FXOKS"
signalling=fxo_ks
callerid="Channel 2" <4002>
mailbox=4002
group=5
context=from-internal
channel => 2
callerid=
mailbox=
group=
context=default
```

```
;;; line="3   WCTDM/4/2
FXSKS"
signalling=fxs_ks
callerid=asreceived
group=0
context=from-pstn
channel => 3
callerid=
group=
context=default
```

```
;;; line="4   WCTDM/4/3
FXSKS"
signalling=fxs_ks
callerid=asreceived
group=0
context=from-pstn
channel => 4
callerid=
group=
context=default
```

### 3.4.3. Configuración de CHAN\_DAHD.CONF

En este archivo, vamos a configurar para Asterisk la interfaz de hardware, en otras palabras aquí configuraremos los canales analógicos.

Para reiniciar el módulo chan\_dahdi escribimos en el CLI *module reload chan\_dahdi.so*.

Configuración final:

```
[channels]
callwaiting=yes
usecallingpres=yes
callwaitingcallerid=yes
threewaycalling=yes
transfer=yes
canpark=yes
cancallforward=yes
```

```

callreturn=yes
echocancel=yes
echocancelwhenbridged=yes
group=1
callgroup=1
pickupgroup=1

```

```

[home-phones]
context=internal ; Uses the [internal] context in extensions.conf
signalling=auto ; fxs_ks Use FXO signalling for an FXS channel - as
set in sytem.conf
dahdichan => 1,2

```

```

[psrn]
context=PSTN ; Incoming calls go to [incoming-psrn-line]
in extensions.conf
signalling=auto ; fxs_ks Use FXS signalling for an FXO channel -
use as set in system.conf
faxdetect=incoming
busydetect=yes
dahdichan => 3,4 ; PSTN attached to port 4

```

### 3.4.4. Configuración de SIP.CONF

type: tipo peer o friend

context: contexto donde entrarán las llamadas generadas.

nat: indica si el usuario o peer se encuentra tras un NAT.

host: IP remota o dynamic (en el caso en que la IP no sea fija).

username: nombre de usuario.

secret: contraseña de acceso en texto plano.

allow/disallow: configuraciones de codecs específicas para cada peer /

friend.

qualify: evalúa el estado del extremo SIP para conocer su accesibilidad y latencia.

canreinvite: permite que el tráfico de voz pase por el asterisk o bien directamente entre las partes.

Creamos las siguientes extensiones sip que van a servir de troncales entre el host físico y las centrales virtualizadas.

Configuración final:

```
[general]  
context=PSTN  
svlookup=no  
videosupport=yes  
allow=all  
t38pt_udptl=yes  
t38pt_rtp=yes  
t38pt_tcp=yes
```

```
allow=gsm  
allow=ulaw  
allow=alaw  
allow=g729  
allow=h263  
allow=h263p  
allow=h264  
canreinvite=no  
allowsubscribe=yes  
notifyringing=yes  
limitonpeer=yes  
notifyhold=yes
```

[7001]  
type=friend  
secret=7001  
qualify=500  
nat=yes  
host=dynamic  
canreinvite=no  
context=tesis  
dtmfmode=rfc2833  
musicclass=default  
allow=all  
call-limit=10  
callgroup=1  
pickupgroup=1

[7002]  
type=friend  
secret=7002  
qualify=500  
nat=yes  
host=dynamic  
canreinvite=no  
context=tesis  
dtmfmode=rfc2833  
musicclass=default  
allow=all  
call-limit=10  
callgroup=1  
pickupgroup=1

[7003]  
type=friend  
secret=7003  
qualify=500  
nat=yes  
host=dynamic  
canreinvite=no  
context=tesis  
dtmfmode=rfc2833  
musicclass=default  
call-limit=20  
callgroup=1  
pickupgroup=1

[7004]  
type=friend  
secret=7004  
qualify=500  
nat=yes  
host=dynamic  
canreinvite=no  
context=tesis  
dtmfmode=rfc2833  
musicclass=default  
subscribecontext=tesis  
call-limit=10  
callgroup=1  
pickupgroup=1

[7005]  
type=friend  
secret=7005  
qualify=500  
nat=yes  
host=dynamic  
canreinvite=no

context=tesis  
dtmfmode=rfc2833  
musicclass=default  
subscribecontext=tesis  
call-limit=10  
callgroup=1  
pickupgroup=1



### 3.4.5. Configuración de EXTENSIONS.CONF

El archivo extensions.conf es la parte central de toda la configuración, dado que es donde se define el dialplan de Asterisk. El dialplan consiste en una serie de contextos que contienen extensiones y sus diferentes prioridades. El archivo tiene los siguientes contextos: general, globals, usuarios-tesis,tesis, outbound-local, outbound-regional, outbound-celular,outbound-internacional, PSTN, además de las macros: voicemail, novicemail, outboundpool

```
[globals]
autofallthrough=no
TIEMPO_MAX=1800000
TIEMPO_ADV=300000
INTERVALO_ADV=60000
OUTBOUNDTRUNK1=DAHDI/3
OUTBOUNDTRUNK2=DAHDI/4

[macro-voicemail]
exten => s,1,Dial(${ARG1},20,rtTL(${TIEMPO_MAX}:${TIEMPO_ADV}:${INTERVALO_ADV})) =>
exten => s,2,VoiceMail(${MACRO_EXTEN}@default,u)
exten => s,3,Hangup()
exten => s,102,VoiceMail(${MACRO_EXTEN},b)
exten => s,103,Hangup()

[macro-novicemail]
exten => s,1,Dial(${ARG1},20,rtTL(${TIEMPO_MAX}:${TIEMPO_ADV}:${INTERVALO_ADV})) =>
exten => s,3,Hangup()

[macro-outboundpool]
```

```

;Implementa la prioridad de salida
exten => s,1,Set(CALLERID(all)="TESIS" <999999>)
exten                                     =>
s,n,Dial(${OUTBOUNDTRUNK1}/${ARG1},,TL(${TIEMPO_MAX}:${TIE
MPO_ADV}:${INTERVALO_ADV}))
exten                                     =>
s,n,Dial(${OUTBOUNDTRUNK2}/${ARG1},,TL(${TIEMPO_MAX}:${TIE
MPO_ADV}:${INTERVALO_ADV}))
exten => s,n,Congestion()
exten => s,n+101,Congestion()
exten => s,n,Hangup()

```

*[usuarios-tesis]*

```

exten => 7000, hint, SIP/7000
exten => 7000, 1, Macro(voicemail, SIP/7000)
exten => 7001, hint, SIP/solgye1
exten => 7001, 1, Macro(novoicemail, SIP/solgye1)
exten => 7002, hint, SIP/solgye2
exten => 7002, 1, Macro(novoicemail, SIP/solgye2)
exten => 7003, hint, SIP/7003
exten => 7003, 1, Macro(novoicemail, SIP/7003)
exten => 7004, hint, SIP/7004
exten => 7004, 1, Macro(novoicemail, SIP/7004)
exten => 7005, hint, SIP/7005
exten => 7005, 1, Dial(DAHDI/2,60,r)

```

*[tesis]*

```

include=usuarios-tesis
include=outbound-local
include=outbound-celular
include=apps
include=outbound-regional
include=outbound-internacional

```

*;Emergencia 911*

```

exten => 911, 1, Dial(${OUTBOUNDTRUNK1}/911)
exten => 911, 2, Dial(${OUTBOUNDTRUNK2}/911)
exten => 911, 3, Congestion()
exten => 911, 103, Congestion()

```

```

exten => 900,1,Dial(SIP/trunk-uio/8007,60,r)
exten => 900,n,Hangup()

```

```
[outbound-local]
```

```
;Llamadas locales
```

```
exten => _NXXXXXX,1,Macro(outboundpool,${EXTEN},60,r)
```

```
;Llamadas a numeros 101,102,etc
```

```
exten => _1XX,1,Macro(outboundpool,${EXTEN})
```

```
;Llamadas a numeros 1-800
```

```
exten => _1800.,1,Macro(outboundpool,${EXTEN})
```

```
exten => _1866XXXXXXX,1,Macro(outboundpool,${EXTEN})
```

```
;Llamadas a numeros 1-700, 1-706, etc
```

```
exten => _170XXXXXXX,1,Macro(outboundpool,${EXTEN})
```

```
[outbound-regional]
```

```
;Llamadas regionales
```

```
;exten => _90[2-7]XXXXXXX,1,Macro(outboundpool-
regional,${EXTEN:1})
```

```
exten => _0[2-7]XXXXXXX,1,Macro(outboundpool,${EXTEN})
```

```
[outbound-celular]
```

```
;Llamadas Celulares
```

```
exten => _0[89]XXXXXXX,1,NoOp(Llamadas a Celular: ${EXTEN})
```

```
exten => _0[89]XXXXXXX,n,Authenticate(19999)
```

```
exten => _0[89]XXXXXXX,n,Macro(outboundpool,${EXTEN})
```

```
[outbound-internacional]
```

```
;Llamadas Internacionales
```

```
exten => _00.,1,NoOp(Llamada Internacional: ${EXTEN})
```

```
exten => _00.,n,Authenticate(9999)
```

```
exten => _00.,n,Dial(SIP/voipcheap/${EXTEN},60,r)
```

```
;exten => _00.,n,Congestion()
```

```
exten => _00.,n+101,Congestion()
```

```
exten => _00.,n,Hangup()
```

```
[apps]
```

```
exten => 200,1,Answer()
```

```
exten => 200,n,VoiceMailMain()
```

```
exten => 200,n,Hangup()
```

```
;Directorio de primer nombre
```

```

exten => 8,1,Answer()
exten => 8,n,Directory(default,tesis,f)
exten => 8,n,Hangup()

```

```

;Directorio de apellido
exten => 9,1,Answer()
exten => 9,n,Directory(default,tesis)
exten => 9,n,Hangup()

```

```

[PSTN]
include=usuarios-tesis

```

## 3.4. INSTALACIÓN DE KVM

### 3.4.1. Preparación

Necesitamos un modelo de procesador y mainboard con soporte para virtualización, para que KVM funcione de forma adecuada. Es posible saberlo examinando `/proc/cpuinfo`. Entonces ejecutamos:

```

grep vmx /proc/cpuinfo          si el procesador es intel
grep svm /proc/cpuinfo          si el procesador es AMD

```

Si se obtienen resultados, el procesador está listo para usar KVM. Si no está seguro de que tipo de procesador tiene, entonces ejecutamos:

```
grep 'vmx|svm' /proc/cpuinfo
```

Si se obtienen resultados con `vmx`, entonces el procesador es Intel, si los obtiene con `svm`, entonces su procesador es AMD. Si no se obtuvieron resultados, entonces el sistema no tiene un CPU construido para la virtualización.

### 3.4.2. Instalando KVM

Hay que configurar los repositorios de yum en la instalación del sistema. Se escribe el siguiente comando para instalar KVM:

```
# yum groupinstall KVM
```

A continuación se instalan los paquetes opcionales y sus dependencias:

```
[root@localhost ~]# yum install celt051-devel etherboot-pxes  
etherboot-roms etherboot-roms-kvm gpxe-roms-qemu iasl kvm-tools  
libcmptutil libvirt-cim qcairo-devel qffmpeg-devel qpixmap-devel qspice
```

*qspice-libs-devel*      *Virtualization-en-US*      *log4cpp-devel*      *SDL-devel.x86\_64*

Después de terminar con la instalación de KVM y antes de poner en funcionamiento las máquinas hay que configurar el acceso a la red de las mismas. Por defecto, KVM viene con NAT desde la máquina donde está instalado KVM. Para poder acceder a las máquinas desde cualquier ubicación hay que configurar un bridge, para que sean accesibles:

### **3.4.3. Configuración Bridge para usar KVM**

Se emplea eth0 para hacer el bridging. Para hacer un bridge en CentOS se necesita hacer 2 pasos.

1. Editar la configuración de red del terminal físico que se va a emplear de bridge y decir que va a servir de bridge y
2. Crear una configuración para un terminal virtual que será el bridge.

Todo esto no sería posible si no estuviera instalado el paquete *bridge-utils*.

En este artículo se ha instalado como dependencia.

Se edita `/etc/sysconfig/network-scripts/ifcfg-eth0`, para eso se ejecuta:

```
# vi /etc/sysconfig/network-scripts/ifcfg-eth0
```

Se actualiza como sigue:

```
DEVICE=eth0
```

```
BOOTPROTO=static
```

```
HWADDR=(La que tenga la tarjeta)
```

```
ONBOOT=yes
```

```
BRIDGE=br0
```

Se guarda y se cierra el archivo. Se crea `/etc/sysconfig/network-scripts/ifcfg-br0`, para eso se ejecuta:

```
# vi /etc/sysconfig/network-scripts/ifcfg-br0
```

En este archivo, se definirán las propiedades del bridge. Se modifica como sigue (hay que tener en cuenta que las opciones son sensibles a mayúsculas y minúsculas por ejemplo `Bridge` y `bridge` son 2 opciones diferentes):

*DEVICE=br0*

*TYPE=Bridge*

*BOOTPROTO=static*

*ONBOOT=yes*

*IPADDR=(Una dirección estática de la red)*

*NETMASK=(Máscara de red, típicamente 255.255.255.0)*

*GATEWAY=(Dirección del router)*

*NETWORK=(Dirección de la red)*

°Se añaden estas líneas a */etc/sysctl.conf* para deshabilitar el filtrado de paquetes en el bridge (si no se hace, es posible que no funcionen ciertas características de las máquinas virtuales que se están ejecutando):

*net.bridge.bridge-nf-call-ip6tables = 0*

*net.bridge.bridge-nf-call-iptables = 0*

*net.bridge.bridge-nf-call-arptables = 0*

Esto mejora el rendimiento del bridge. Se recomienda el uso de filtrado de paquetes en los ordenadores que se conectan a través del bridge, pero no en el propio bridge.



Aplicamos los cambios de sysctl:

```
sysctl -p /etc/sysctl.conf
```

Reiniciamos los interfaces de red: `service network restart`

Una vez que se tiene configurado el Bridge y antes de empezar a crear o a emplear máquinas virtuales, hay que reiniciar la máquina para que coja la instalación de KVM.

**CAPITULO IV**

**FUNCIONAMIENTO Y PRUEBAS**

## 4.1. Iniciando Virtual Machine Manager

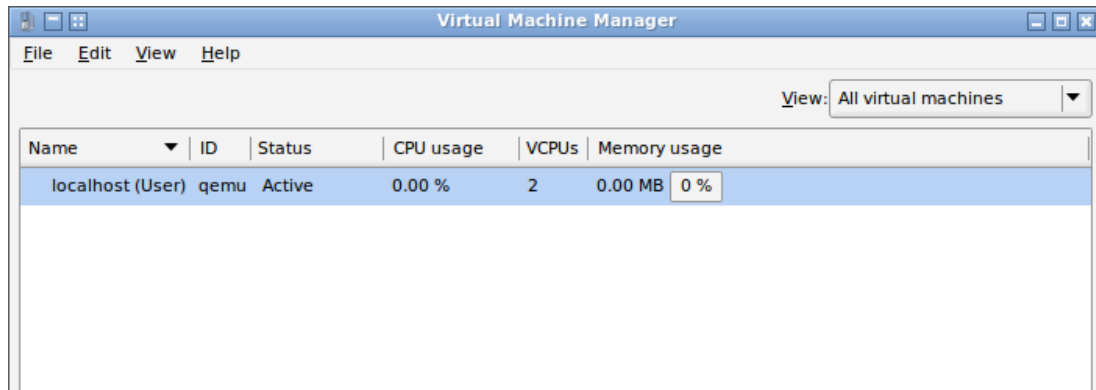


Fig. 4.1 Virtual Machine Manager

Comenzamos por el lanzamiento de virt-manager (fig. 4.1) desde la interfaz gráfica seleccionando *Aplicaciones -> Herramientas del sistema -> Virtual Machine Manager*, o desde la línea de comandos en una ventana de terminal mediante la ejecución del comando `su - /sbin /service libvirtd inicio`.

Una vez cargado, el administrador de la máquina virtual le pedirá la contraseña del root antes de mostrar la pantalla del Virtual Machine Manager.

La pantalla principal virt-manager lista las máquinas actuales virtuales configuradas en el sistema. En este momento sólo debe haber una, el sistema host. Por defecto, el manager debe estar conectado a la máquina. Si no lo está, se conectan al sistema del host haciendo clic derecho sobre la entrada en la lista y seleccionamos *Conectar* en el menú que nos muestra.

## 4.2. Creando una máquina virtual



Fig. 4.2 Creando un nuevo sistema virtual

Para crear un sistema virtual nuevo, nos aseguramos de que el host está seleccionado en la lista y hacemos clic en el botón *Nuevo* para mostrar la primera pantalla del asistente *Crear una nueva máquina virtual* (fig. 4.2). Leemos la información en la primera pantalla y a continuación, hacemos clic en el botón *Adelante* para continuar.

En el campo Nombre en la siguiente pantalla, escribimos un nombre descriptivo adecuado para el sistema virtual, en nuestro caso *VirtualAsterisk-1* (fig. 4.3)

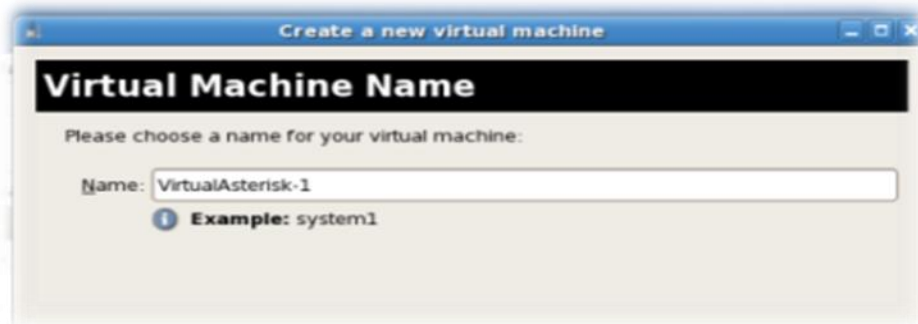


Fig. 4.3 Creando un nuevo sistema virtual I

En la siguiente pantalla, seleccionamos el método de virtualización que se utilizará para la máquina virtual. Dado que el plan para ejecutar Elastix, usaremos virtualización completa. También seleccionamos *KVM* en el menú *Hypervisor* y la arquitectura de CPU para el huésped (i686 de 32 bits o *x86\_64* para 64 bits) en nuestro caso *x86\_64*.

Tenga en cuenta que si la opción *KVM* no está disponible en el menú *hypervisor* puede ser que *KVM* no se ha instalado correctamente en el sistema.

#### 4.2.1. Configurando el Método de Instalación

En la siguiente pantalla, seleccionamos el método de instalación. Las opciones soportadas incluyen los medios de comunicación locales, tales como un CD-ROM, DVD o archivo de imagen ISO, una instalación de red a través de HTTP, FTP o NFS o una instalación de inicio de red basada en PXE. En esta pantalla, también se especifica la familia de sistemas operativos y la versión del guest.

Entonces escogeremos para la instalación desde la unidad de CD-Room *Local Install Media*, la opción para *Linux* en el tipo de sistema operativo, y en variante escogemos *Red Hat Enterprise Linux 5.4 or later*.

En la siguiente pantalla, configuramos los parámetros correspondientes a la selección del medio de instalación realizado en la pantalla anterior (dispositivo de CD-ROM) en nuestro caso Elastix y hacemos clic en *Siguiente* para proceder.

#### **4.2.2. Configurando el almacenamiento virtual KVM**

El *guest* requiere espacio en disco para almacenar el sistema operativo y los archivos de datos de usuario. La siguiente pantalla del asistente de instalación permite configurar estas opciones de almacenamiento. Al *guest* se le puede asignar una partición de disco o un archivo de imagen que reside en el sistema de archivos del host para utilizarlo como almacenamiento.

En nuestro caso, asignaremos *35000 Mb* de tamaño de disco y guardaremos la imagen de la máquina virtual en */var/lib/xen/images/VirtualAsterisk1.img*



Fig. 4.4 Almacenamiento en KVM

Una vez realizados estos ajustes, hacemos clic en *Adelante* para configurar parámetros de red del sistema operativo guest. (fig. 4.4).

### 4.2.3. Configurando las opciones de redes KVM

Varias opciones están disponibles para proporcionar conectividad de red para un equipo virtual basado en KVM, como lo muestra la fig. 4.5. Si el cliente se conecta a una red virtual en el sistema operativo host con

conectividad proporcionada a la red externa a través de Network Address Translation (NAT), entonces la red se debe seleccionar la opción virtual.

En nuestro caso, la máquina virtual se conecta directamente a la red externa mediante el intercambio de un adaptador de red instalado en el host, entonces la opción de dispositivo físico compartido debe ser seleccionado. Debemos tener en cuenta que la opción de dispositivo para compartir, requiere que un puente de red ya se puede configurar en el host como se describimos en el capítulo 3.4.3.



Fig. 4.5 Red en KVM



#### **4.2.4. Configuración de las opciones de memoria de la máquina virtual KVM y la CPU**

La nueva máquina virtual tendrá acceso a los recursos de la CPU y la memoria antes de que pueda ser iniciado. Estas opciones se configuran en el *Asistente de asignación de memoria y CPU*.

La configuración óptima dependerá del número de CPU y la cantidad de memoria física presente en el host y los requisitos de otras aplicaciones y máquinas virtuales que corren en paralelo con la nueva máquina virtual. En términos de asignación de memoria, una cantidad de arranque se puede especificar junto con la memoria máxima que se puede asignar en tiempo de ejecución de la máquina virtual:

Para nuestras máquinas virtuales, asignaremos 300 Mb de memoria y una unidad de CPU respectivamente. (fig.4.6)

Al hacer clic en el botón *Forward* después de realizar la configuración de la memoria y la CPU se mostrará una pantalla de resumen. Debemos asegurarnos de revisar que la información coincida con las expectativas para la máquina virtual antes de proceder.

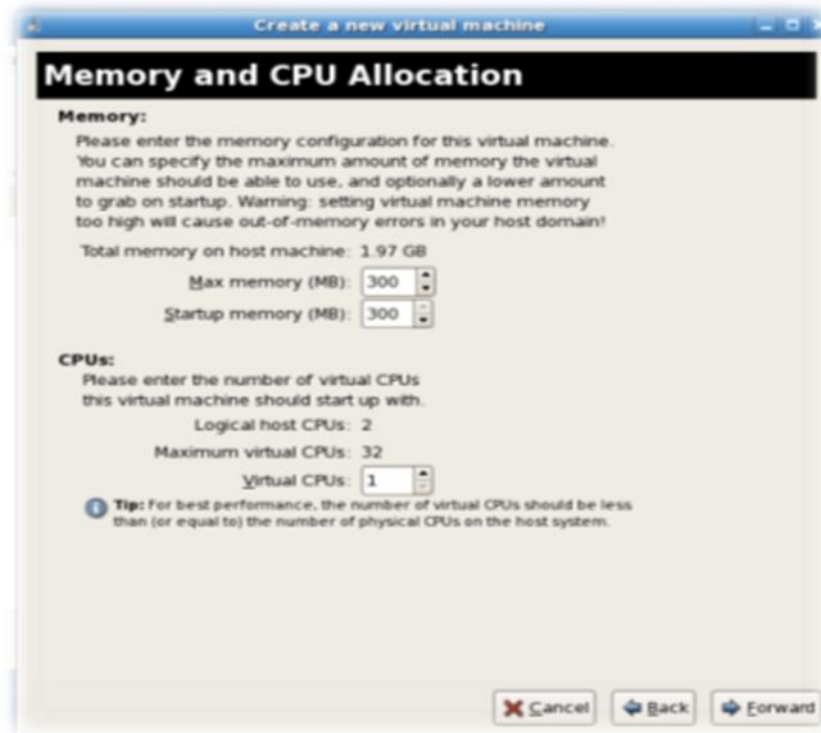


Fig. 4.6 Memoria y CPU en KVM

### 4.3. Inicializando la nueva máquina virtual

Hacemos clic en el botón *Finalizar* para iniciar el proceso de creación. El administrador de virtualización creará el disco y configurará la máquina virtual antes de iniciar el sistema. La duración de este proceso dependerá de varios factores incluyendo el tamaño del disco asignado a la máquina virtual, si el disco se ha preasignado y la ubicación de los medios de instalación. Una vez que el proceso de creación se completa, la nueva máquina virtual aparecerá en la ventana principal Virtual Machine Manager.

De forma predeterminada, la consola de la máquina virtual debe aparecer en la ventana de la Virtual Machine Viewer. Para ver la consola de la máquina en marcha en cualquier momento futuro, asegúrese de que está seleccionada en la lista de máquinas virtuales y seleccione el botón Abrir de la barra de herramientas inferior. El visor de la máquina virtual debe estar listo para iniciar el proceso de instalación. A partir de ahora, simplemente seguimos las instrucciones de instalación de Elastix en la máquina virtual KVM y listo. (fig. 4.7)



```
- To install or upgrade in graphical mode, press the <ENTER> key.  
- To install or upgrade in text mode, type: linux text <ENTER>.  
- Use the function keys listed below for more information.  
[F1-Main] [F2-Options] [F3-General] [F4-Kernel] [F5-Rescue]  
boot: _
```

**Fig. 4.7 Instalando Elastix en la nueva VM**

#### 4.4. CONFIGURANDO ELASTIX

Tendremos en cuenta los siguientes parámetros para la configuración del plan de marcado de Elastix:

##### VIRTUAL1

**Dirección IP: 192.168.0.195**

*EXTENSIONS IAX TRUNK*  
1000= SERVIDOR VIRTUAL2  
1001= SERVIDOR VIRTUAL3  
*SIP*  
1500 SIPVIRTUAL1

##### VIRTUAL2

**Dirección IP: 192.168.0.156**

*EXTENSIONS IAX TRUNK*  
2000 SERVIDOR VIRTUAL1  
2001 SERVIDOR VIRTUAL3  
*SIP*  
2500 SIPVIRTUAL2

##### VIRTUAL3

**Dirección IP: 192.168.0.158**

*EXTENSIONS IAX TRUNK*  
3000 SERVIDOR VIRTUAL 2  
3001 SERVIDOR VIRTUAL1  
*SIP*  
3500 SIPVIRTUAL3

A continuación vamos a configurar el Trunk Sip para cada máquina virtual desde *PBX- PBX Configuration- Trunks – Sip Trunk*

#### 4.4.1. Configuración del Trunk Sip

Las máquinas virtuales se están conectando al host principal a través de las troncales SIP:

##### Máquina virtual 1

*Trunk description: DAHDI SIP*  
*Outbound Caller Id:*  
*Cid Options: Allow any CID*

##### OUTGOING DETAILS

###### PEER Details

*host= 192.168.0.153*  
*username= 7001*  
*secret = 7001*  
*type= friend*

##### REGISTRATION

*Register String:*  
*7001:7001@192.168.0.153/7001*

##### Máquina virtual 2

*Trunk description: DAHDI SIP*  
*Outbound Caller Id:*  
*Cid Options: Allow any CID*

##### OUTGOING DETAILS

###### PEER Details

*host= 192.168.0.153*  
*username= 7002*

```
secret = 7002  
type = friend
```

#### REGISTRATION

```
Register String:  
7002:7002@192.168.0.153/7002
```

### Máquina virtual 3

```
Trunk description: DAHDI SIP  
Outbound Caller Id:  
Cid Options: Allow any CID
```

#### OUTGOING DETAILS

```
PEER Details  
host = 192.168.0.153  
username = 7003  
secret = 7003  
type = friend
```

#### REGISTRATION

```
Register String:  
7003:7003@192.168.0.153/7003
```

Como podemos observar, repetimos el mismo procedimiento con las otras dos máquinas virtuales con los usuarios 7002 y 7003 respectivamente. Luego para confirmar el registro de las extensiones ejecutamos en el CLI de asterisk en el servidor físico el comando *sip showpeers* (Fig. 4.8).

```
sip showpeers
```

```

peer peer
VIRTUAL*CLI> sip show peers
Name/username      Host                Dyn Forceport ACL Port      Status
7001/7001          192.168.0.247      D   N         5310     OK (10 ms)
7002/7002          192.168.0.156      D   N         5060     OK (32 ms)
7003              (Unspecified)      D   N         0        UNKNOWN
7004              (Unspecified)      D   N         0        UNKNOWN
7005              (Unspecified)      D   N         0        UNKNOWN
5 sip peers (Monitored: 2 online, 3 offline Unmonitored: 0 online, 0 offline)
[Feb 18 17:45:48] NOTICE[2400]: chan_sip.c:23343 handle_request_subscribe: Received SIP subscribe for peer without
-- Unregistered SIP '7001'
== Using UDPTL CoS mark 5
== Using SIP SIP CoS mark 5
-- Executing [7005@sofia/1] Dial("SIP/7002-00000001", "DAHDI/2,60,s") in new stack
-- Called 2
-- DAHDI/2-1 is ringing
-- DAHDI/2-1 is ringing
-- DAHDI/2-1 is ringing
-- DAHDI/2-1 is ringing
-- DAHDI/2-1 answered SIP/7002-00000001
-- Hanging up on 'DAHDI/2-1'
-- Hungup 'DAHDI/2-1'
== Spawn extension (sofia, 7005, 1) exited non-zero on 'SIP/7002-00000001'
-- Starting simple switch on 'DAHDI/2-1'
-- Hanging up on 'DAHDI/2-1'
-- Hungup 'DAHDI/2-1'
-- Registered SIP '7001' at 192.168.0.195:5060
> Saved useragent "Asterisk PBX 1.6.2.13" for peer 7001
[Feb 18 17:02:52] NOTICE[2400]: chan_sip.c:19580 handle_response_peerpoke: Peer '7001' is now reachable. (6ms / 50
VIRTUAL*CLI>

```

Fig. 4.8 Salida del SIP SHOW PEERS

#### 4.4.2. Configuración del Trunk IAX2

El objetivo de agregar este trunk es comunicar las máquinas virtuales entre sí. A continuación detallamos los parámetros configuraremos desde la interfaz web de Elastix, en la opción *ADD IAX2 TRUNK*:

#### DESDE LA MÁQUINA VIRTUAL 1

##### *IAX2 TRUNK*

##### *General Settings*

*Trunk description:* Virtual2

*CID options:* allow any CID

##### *Outgoing Settings*

##### *PEER Details:*

*Host=* 192.168.0.156

*Username=* 1000

*Secret=* 1000

*Type=* peer

*Trunk= yes*

*Incoming Setting*

*Secret= \*\*\*password\*\*\**

*Type=user*

*Context= from-trunk*

*IAX2 TRUNK*

*General Settings*

*Trunk description: Virtual3*

*CID options: allow any CID*

*Outgoing Settings*

*PEER Details:*

*Host= 192.168.0.158*

*Username= 1001*

*Secret= 1001*

*Type= peer*

*Trunk= yes*

*Incoming Setting*

*Secret= \*\*\*password\*\*\**

*Type=user*

*Context= from-trunk*

## **DESDE LA MÁQUINA VIRTUAL 2**

*IAX2 TRUNK*

*General Settings*

*Trunk description: Virtual1*

*CID options: allow any CID*

*Outgoing Settings*

*PEER Details:*

*Host= 192.168.0.195*

*Username= 2000*

*Secret= 2000*

*Type= peer*



*Trunk= yes*

*Incoming Setting*

*Secret= \*\*\*password\*\*\**

*Type=user*

*Context= from-trunk*

### **DESDE LA MÁQUINA VIRTUAL 3**

*IAX2 TRUNK*

*General Settings*

*Trunk description: Virtual1*

*CID options: allow any CID*

*Outgoing Settings*

*PEER Details:*

*Host= 192.168.0.195*

*Username= 3000*

*Secret= 3000*

*Type= peer*

*Trunk= yes*

*Incoming Setting*

*Secret= \*\*\*password\*\*\**

*Type=user*

*Context= from-trunk*

*IAX2 TRUNK*

*General Settings*

*Trunk description: Virtual2*

*CID options: allow any CID*

*Outgoing Settings*

*PEER Details:*

*Host= 192.168.0.156*

*Username= 3001*

*Secret= 3001*

*Type= peer*

*Trunk= yes*

```
Incoming Setting  
Secret= ***password***  
Type=user  
Context= from-trunk
```

Por último, configuramos las rutas salientes por medio de ADD ROUTE de la siguiente manera:

```
Route Name : virtual1  
Dial Patterns : 1.  
Trunk Sequence: IAX2/1000
```

#### 4.5. CONFIGURACIÓN DEL TELÉFONO IP CISCO SPA 303

Primero obtenemos la dirección Ip del teléfono, en nuestro caso 192.168.0.110

Luego, en el pc, abrimos un navegador donde digitaremos la IP en mención.

En la interface web nos cargaran tres pestañas principales:

- Voice – Contiene información acerca del teléfono, incluye los siguientes subpestañas:
  - Info— Contiene información del sistema y del teléfono (sólo lectura).
  - System—Permite asignar la contraseña y configuraciones de red.
  - Phone—Permite habilitar el screen saver (Cisco SPA 303).
  - User— Contiene configuraciones para call forwarding, speed dials, call waiting, configuraciones de privacidad y audio.

- Call History—Muestra las llamadas recibidas y hechas desde el teléfono.
- Personal Directory— Contiene la libreta de direcciones.

Ahora procedemos a configurar las extensiones SIP del teléfono:

*Ext. 1*

*General*

*Line Enable = yes*

*Share Line Appearance*

*Share Ext. =private*

*Subscription Expires = 3600*

*NAT Setting*

*NAT Mapping Enable = no*

*NAT Keep Alive Msg = SNOTIFY*

## **4.6. PRUEBAS**

Para efectos de pruebas de estabilidad y escalabilidad de nuestro sistema, hemos optado por usar SIPP, una herramienta gratuita de código abierto que genera tráfico para el protocolo SIP y que además muestra dinámicamente estadísticas sobre las pruebas ejecutadas.

A continuación presentamos los resultados de generados por esta herramienta en nuestro sistema:

Con el test realizado con el software SIPP, se definió que una central IP con nuestras características puede albergar unas 316 llamadas concurrentes, de las cuales realizó 1265 llamadas en un lapso de 126.51 segundos (fig. 4.8).

```

Call-rate(length)  Port  Total-time  Total-calls  Remote-host
10.0(10000 ms)/1.000s  5061  126.51 s  1265  192.168.0.195:5060(UDP)

10 new calls during 1.002 s period  4 ms scheduler resolution
316 calls (limit 1000)  Peak was 316 calls, after 31 s
0 Running, 646 Paused, 82 Woken up
0 dead call msg (discarded)  0 out-of-call msg (discarded)
3 open sockets

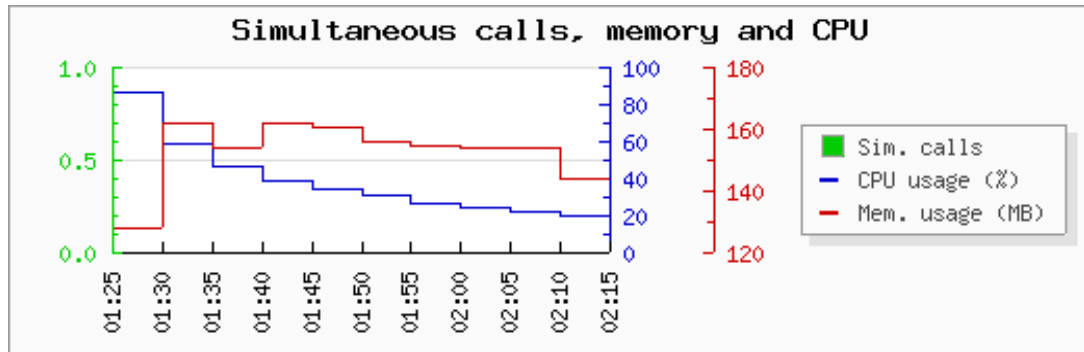
Messages  Retrans  Timeout  Unexpected-Msg
INVITE ----->  1265  6038  949
100 <-----  0  0  0  0
180 <-----  0  0  0  0
183 <-----  0  0  0  0
200 <----- E-RTD1  0  0  0  0
ACK ----->  0  0
Pause [ 10.0s]  0
BYE ----->  0  0  0  0
200 <-----  0  0  0  0

----- [ + | - | * | / ] : Adjust rate ---- [ q ] : Soft exit ---- [ p ] : Pause traffic -----
Last Error: Aborting call on UDP retransmission timeout for Call-ID '949...'

```

Fig. 4.9 Testing con SIPP

Este gráfico extraído de la pestaña *Sistemas* del administrador web de Elastix, nos muestra cómo el uso del cpu y la memoria aumentaron casi a la par llegando a un pico del 80% y 75% de uso respectivamente durante proceso del testing (fig. 4.9).



**Fig. 5** Uso de Memoria y CPU

## CONCLUSIONES

Como conclusión del trabajo realizado podemos anotar que:

1. 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 maquina. En esta nueva era, la virtualización se está imponiendo con fuerza debido a que hoy en día muchos procesos dependen de la capacidad que tengan de hacer tareas múltiples, tomándolo como opción incluso para combatir la crisis económica actual.
2. La virtualización es una herramienta muy importante para el cuidado del ambiente, porque no se utiliza mucha energía, se ahorra espacio físico y los equipos de enfriamiento no deben ser muy potentes porque se utiliza menos hardware
3. Con el hardware utilizado en las pruebas, se pudo instalar y correr unas 6 PBX Virtuales, teniendo un total de 1.20 GB utilizados y dejando 800 MB al servidor de virtualización para su funcionamiento, el Procesador llego al pico más alto cuando se apagaban o encendían, de ahí se mantenía en uso alrededor del 40% de la capacidad total.

## RECOMENDACIONES

Dadas las conclusiones antes descritas, podemos recomendar lo siguiente:

1. Cuando se vaya a elegir un software de virtualización de ordenadores, es importante realizar un estudio de la infraestructura hardware a nivel de servidores con que cuenta su organización; con el fin de definir parámetros determinantes en la elección del software de virtualización de ordenadores.
2. Para realizar pruebas, experimentos e implementaciones se recomienda trabajar con software de virtualización de ordenadores, en plataformas de distribución libre, ya que se basan en paquetes bien estables y ajustados por distros de Linux.
3. Es mejor implementar soluciones de virtualización en KVM puesto que puede implementarse PARAVIRTUALIZACIÓN o VIRTUALIZACIÓN COMPLETA, mientras que con similares como VIRTUALBOX o VMWARE SERVER solo puede disfrutarse de una Virtualización por Software, la cual tiene un rendimiento inadecuado para entornos de producción.

4. Para tener un mejor rendimiento del servidor de virtualización dejarle 1.5 GB al host físico aparte de la memoria utilizada en los hosts virtuales.
  
5. Se recomienda utilizar discos SAS por su velocidad de lectura y escritura.



## GLOSARIO DE TERMINOS

***Bare-metal:*** El término bare-metal se refiere a la arquitectura física subyacente de un ordenador. Ejecución de un sistema operativo en el bare-metal es otra forma de referirse a la ejecución de una versión modificada del sistema operativo en el hardware físico.

***dom0:*** También conocido como host o sistema operativo anfitrión. dom0 se refiere a la instancia host de Xen corriendo sobre el hipervisor el cual facilita la virtualización de sistemas operativos invitados.

***Full virtualization:*** Puede implementar virtualización de CentOS en una de dos opciones: virtualización completa o para-virtualización. La virtualización completa proporciona total abstracción del sistema físico subyacente (bare-metal) y crea un nuevo sistema virtual en que los sistemas operativos el invitados puede ejecutarse. No son necesarias modificaciones en el sistema operativo invitado. Los sistemas operativos invitados y las aplicaciones en los invitados no tienen conocimiento del ambiente de virtualización y se ejecutan normalmente. Paravirtualización requiere una versión modificada del operativo Linux sistema.

**Hypervisor:** El hipervisor es la capa de software que abstrae el hardware desde el sistema operativo permitiendo múltiples sistemas operativos funcionar en el mismo hardware.

**Host:** El sistema operativo host, también conocido como dom0.

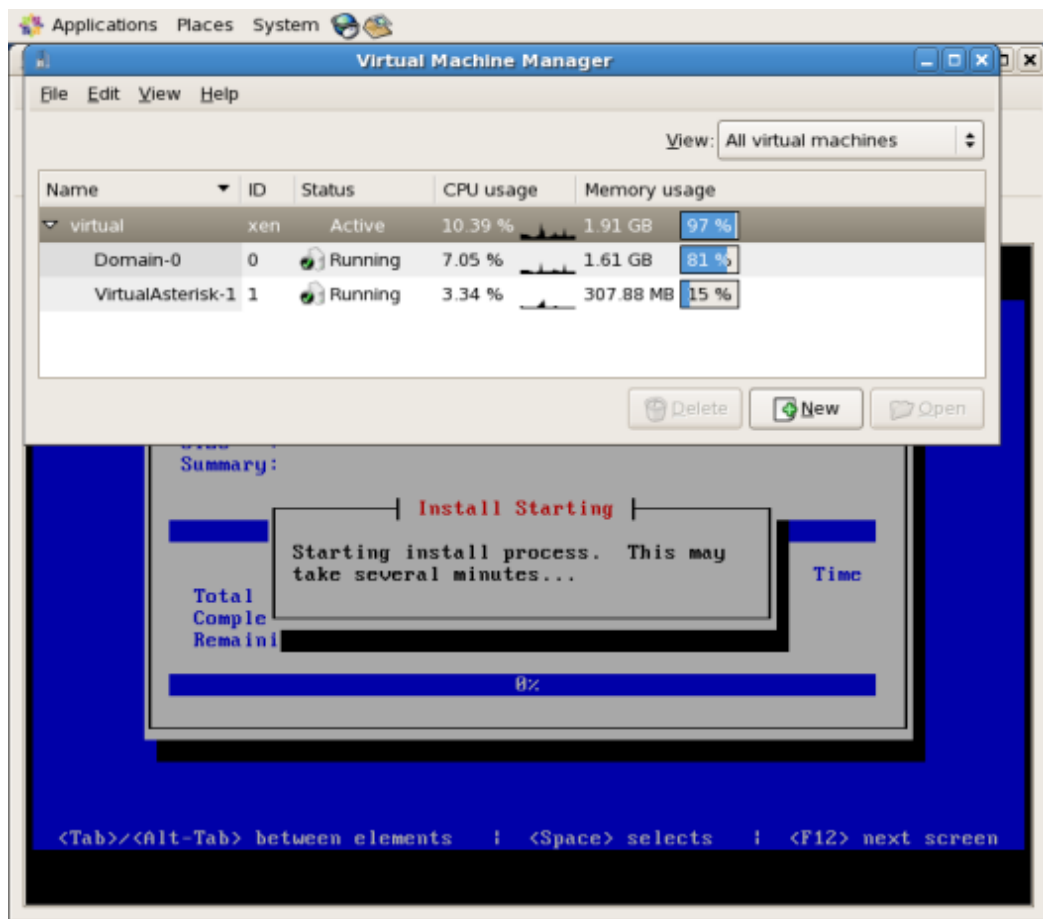
**Kernel-based Virtual Machine:** KVM es un módulo del kernel de virtualización completa que se incorporará en futuras versiones de CentOS. KVM es en la actualidad disponible en la distribución de Linux Fedora y otras distribuciones de Linux.

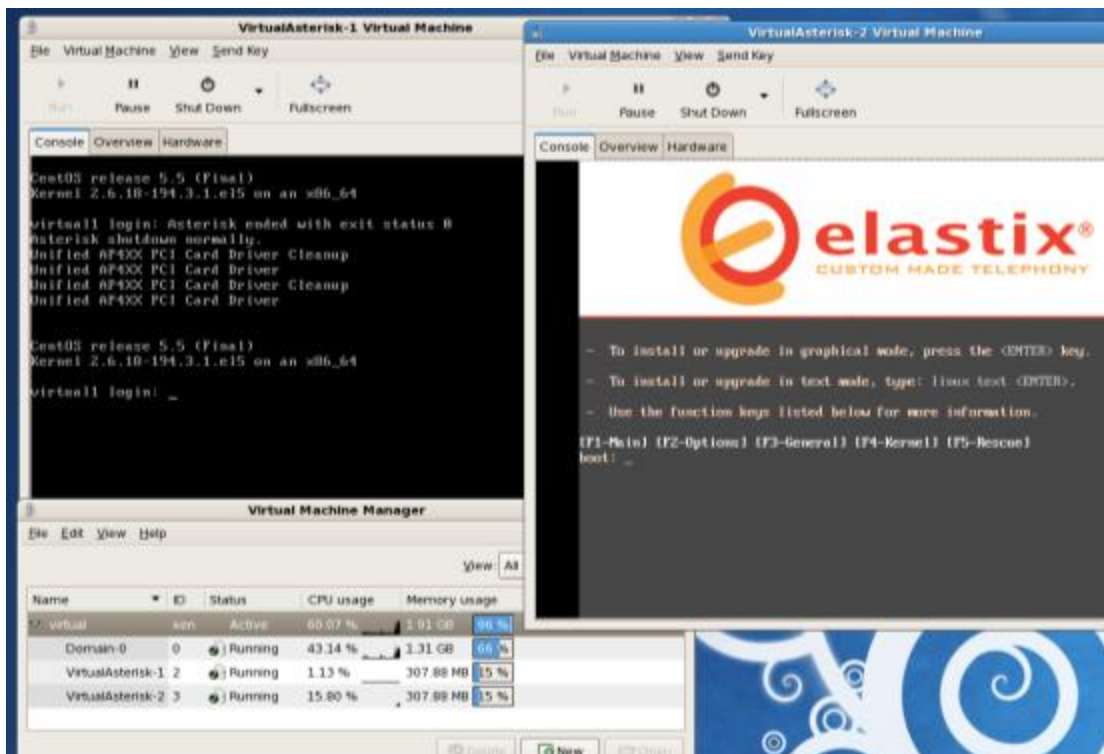
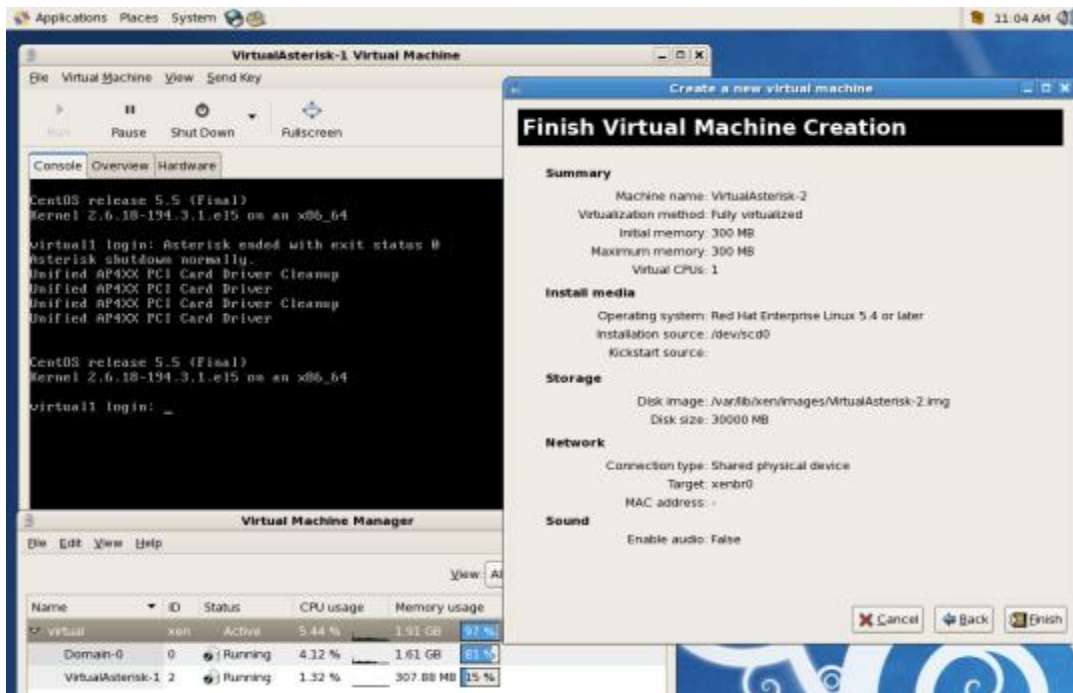
**Migración:** La migración se refiere al proceso de traslado de un para-virtualizados clientes imágenes de un servidor de virtualización de Red Hat a otro. Este otro el servidor podría estar en el mismo servidor o un servidor diferente, incluyendo servidores en otras ubicaciones.

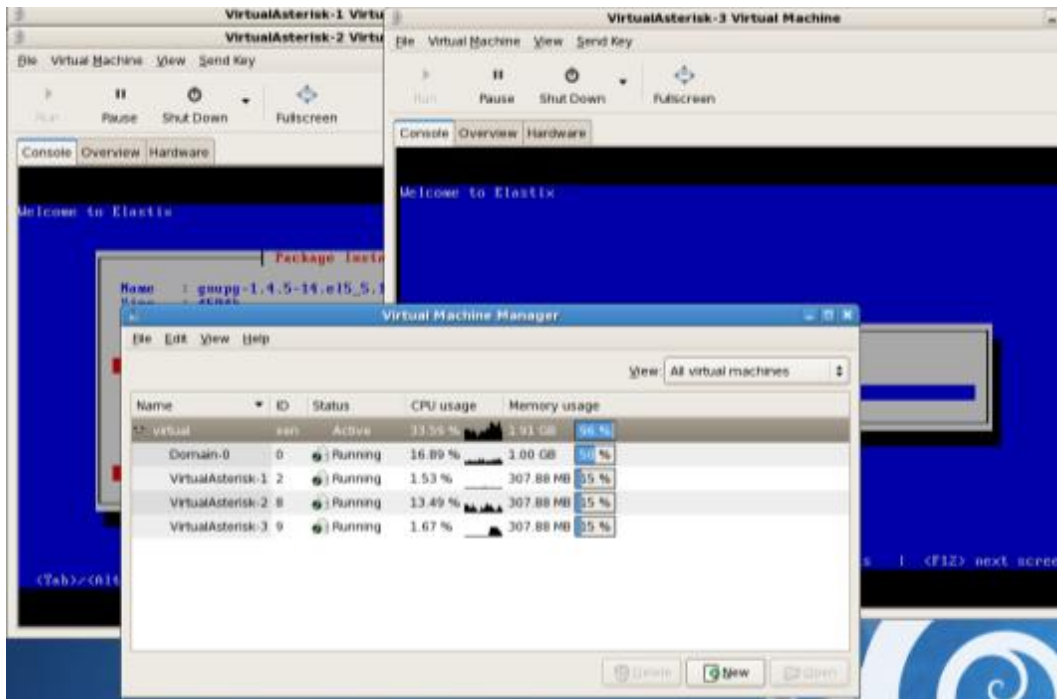
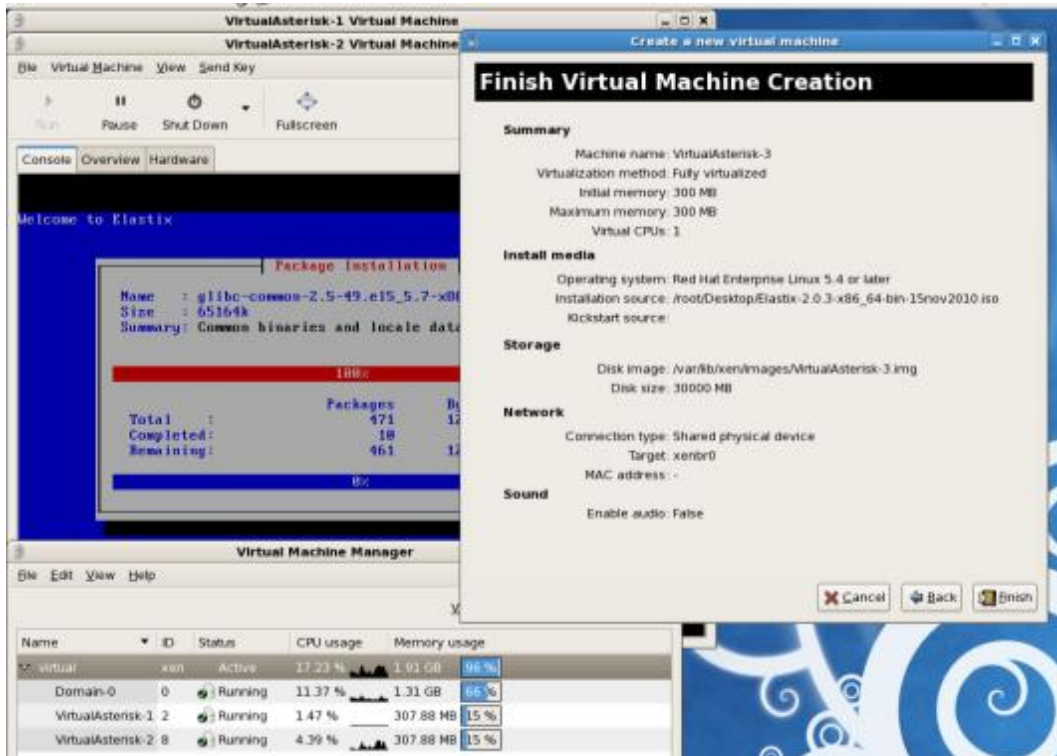
**Para-virtualization:** Para-virtualización utiliza un núcleo especial, a veces se denomina el núcleo xen kernel o kernel-xen xen para virtualizar otro entorno, mientras se utiliza las librerías de ordenadores y dispositivos.

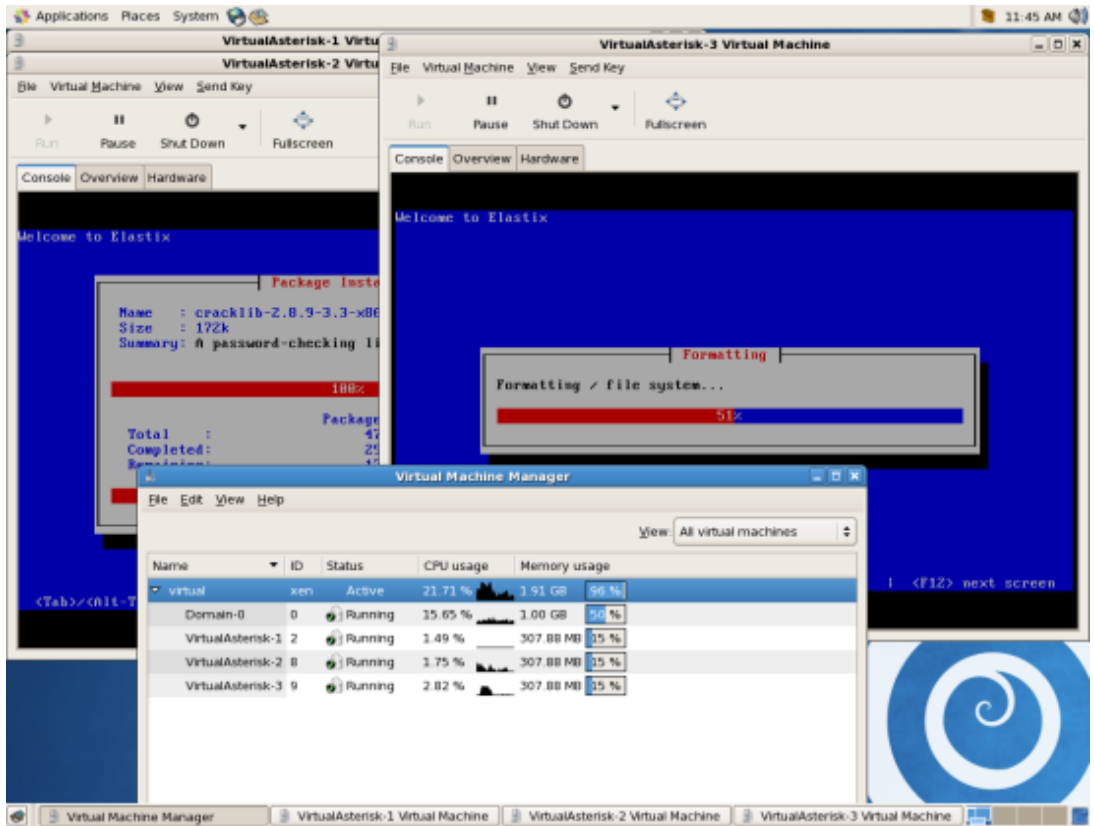
# ANEXOS

## RENDIMIENTO DURANTE EL PROCESO DE INSTALACIÓN DE ELASTIX EN LAS VM









## BIBLIOGRAFÍA

- 1.- Muy Pymes, “Cómo funciona la virtualización y para qué sirve”,  
<http://www.muypymes.com/tecnologia/software/4605-como-funciona-la-virtualizacion-y-para-que-sirve.html>, Octubre del 2009
- 2.- George Gilder, “Las fábricas de la información”,  
[http://www.wired.com/wired/archive/14.10/cloudware.html?pg=1&topic=cloudware&topic\\_set=](http://www.wired.com/wired/archive/14.10/cloudware.html?pg=1&topic=cloudware&topic_set=), Octubre del 2006
- 3.- Salesforce.com Spain S.L, “Cloud Computing según Salesforce”,  
<http://www.salesforce.com/es/cloudcomputing/>, 2010
- 4.- Movistar, “Cloud Computing”, <http://www.mcloud.cl/>, 2010
- 5.- Zeroth, “Concepto general de la Wikipedia sobre el Cloud Computing”,  
[http://es.wikipedia.org/wiki/Computaci3n\\_en\\_nube](http://es.wikipedia.org/wiki/Computaci3n_en_nube), 11 de Febrero del 2011.
- 6.- Andrea Cummins, “Ventajas y desventajas de la computación en nube”,  
<http://geeksroom.com/2010/04/16293/16293>, 14 de Abril del 2010
- 7.- Víctor Fernández y Javier Leyton, “Aplicando el Cloud Computing”,  
<http://profesores.elo.utfsm.cl/~agv/elo322/1s10/project/reports/cloudcomputing-10s01.pdf>, 14 de Julio del 2010.

- 8.- Tango/04 Computing Group, “Referencia de acuerdos de SLA para los Data Center en España”,  
<http://www.barcelona04.com/publicdocs/SLA.doc>, Mayo del 2001
- 9.- Giovanni Silva, “Centro de Datos de Alta Disponibilidad”,  
[http://www.amereiaf.org.mx/congreso2008/materiales/experiencias/Datacenter\\_dealta\\_disponibilidad.pdf](http://www.amereiaf.org.mx/congreso2008/materiales/experiencias/Datacenter_dealta_disponibilidad.pdf), 27 de Noviembre del 2008