Secretos del VM: Virtualización y Drivers

Kathryn M. Jones Pérez
Universidad de Costa Rica, Dep. de Ingeniería,
San José, Costa Rica
kathrynster@gmail.com

y

Esteban González
Universidad de Costa Rica, Dep. de Ingeniería,
San José, Costa Rica
egon256@gmail.com

Abstract:

Virtualization is a program designed to allow the user to have functionality a normal computer could never provide. For example, use of more than one operating system at the same time on the same computer without affecting the computers main operating system. This software can be run on any computer with all the normal qualities. This also allows the utilization of any hardware. The equipment can be partially damaged or it can physically be on a different computer.

Key words: Virtual Machine, virtualization, Virtual Private Network (VPN)

Resumen:

La virtualización consiste en programas diseñados con la función de permitir al usuario capacidades que su computadora normalmente no tendría. Por ejemplo, utilizar más de un sistema operativo al mismo tiempo sin afectar el sistema operativo principal usado en la computadora. Dicho software tiene la capacidad de poder ejecutarse sobre cualquier ordenador. Además, permite el aprovechamiento de cualquier hardware ya sea un equipo parcialmente dañado o equipo repartido en varias maquinas físicas.

Palabras clave: Máquina Virtual, virtualización, Virtual Private Network (VPN)

1 Introducción

En la era de la informática siempre ha sido de crucial importancia el manejo eficiente de recursos computacionales. En muchas ocasiones se llega a tener un gran número de computadores y equipo de red para suplir las necesidades de una empresa, y administrar todo ese equipo y además el espacio para almacenarlo es un trabajo apremiante.

Pero, ¿qué tal sería si pudiéramos fusionar varios computadores en uno sólo? ¿O poder administrar toda una granja de servidores desde una sola terminal? Gracias a la virtualización, esto es posible. Con esta tecnología es posible abstraer recursos computacionales de manera que podemos tener computadores virtuales con todas las funcionalidades de un computador real.

La virtualización es una herramienta poderosa para el aprovechamiento de recursos, y hasta sustituto de algunos de ellos. El manejo interno de los mismos es complejo y presenta obstáculos como lo son la problemática de los drivers. Pero mediante el manejo adecuado, logra sacar lo mejor de esta tecnología sin dificultades.

Esta investigación pretende exponer al lector las generalidades de la virtualización de recursos computacionales, dando un enfoque en las maquinas virtuales y redes virtuales (VPN), su función, instalación y procesos. El documento incluye una explicación de las maquinas virtuales, su origen, interacción con el sistema operativo, y estructura básica mediante un ejemplo. Luego se expone otra aplicación de la virtualización que es la virtualización de recursos, más específicamente una red virtual. Seguido de una breve explicación de drivers, se presenta los principales problemas de los drivers y algunas soluciones propuestas para su manejo. Después se describirán algunas amenazas de seguridad que presenta la virtualización. Finalmente se presentan las conclusiones de la investigación.

2 ¿Qué es Virtualización?

Una definición de virtualización es:"aquello que tiene una existencia aparente y no es real." [1] En informática, virtualización es un término amplio que se refiere a la abstracción de los recursos de una computadora. La meta principal detrás de tecnologías de virtualización es la de ocultar los detalles técnicos a través de la encapsulación de los mismos. La virtualización crea un interfaz externa que oculta una implementación más compleja mediante la mezcla de recursos en lugares físicas diferentes, o mediante la simplificación del sistema de control. [11]

En general, la virtualización produce un ahorro grande de costos en muchos sectores de la computación, sobre todo en el área de tecnologías de información, donde con frecuencia es necesario dar mantenimiento a granjas de servidores de la manera más eficiente posible. Tener pocas máquinas físicas, ampliadas con las máquinas virtuales es mucho más fácil y barato de mantener que la alternativa de tener un servidor físico para cada solución necesaria. Además la virtualización ofrece ventajas muy deseables en este tipo de situaciones.

Existen dos formas de virtualización: virtualización completa y paravirtualización. La virtualización completa consiste en una instancia completa de una computadora. Para lograr una virtualización completa real, todas las operaciones del CPU deben ser reproducidas en un procesador virtual. Pero el retardo de reproducir cada instrucción hace que esta técnica sea poco práctica o hasta imposible. El uso de una máquina virtual (las cuales se explicarán con más detalle más adelante) provee una representación suficientemente cercana, sin el retardo, para ser considerado una virtualización completa. Se logra un mayor acercamiento a virtualización completa si se cuenta con hardware virtualizado el cual provee funcionalidad específica para un sistema virtualizado. [18]

La paravirtualización logra mayor rendimiento sobre la virtualización completa mediante una modificación al sistema operativo (SO) invitado. Los SO invitados o virtualizados "saben" que son virtuales, por lo que se reducen los cambios de contexto. La interacción entre drivers depende de los drivers del anfitrión. Es decir comparten la funcionalidad de los drivers. El anfitrión es el encargado de coordinar el uso del driver. [18] Paravirtualización no se explicará en este artículo.

3. Virtualización completa

Como se mencionó anteriormente, la virtualización completa es la creación de una computadora completa dentro de un equipo anfitrión. La manera en que esto se logra es utilizando máquinas virtuales. A continuación se explicarán los detalles del concepto, origen, interacción con el sistema operativo y se explicará la estructura básica mediante un ejemplo real.

3.1 Maquina Virtual

La idea central de una maquina virtual es permitir ejecutar varios sistemas operativos en un sólo hardware [1]. Las maquinas virtuales ofrecen una manera de aprovechar mejor los recursos disponibles. La desventaja de esto es que si se daña el hardware, todos los sistemas que operaban en ese equipo también fallan, pero la solución fácil es que una maquina virtual se puede trasladar a otro equipo fácilmente.

Una máquina virtual es un sistema sencillo, utilizado en muchos dispositivos electrónicos. Se puede definir específicamente como: "Un duplicado eficiente y aislado de una máquina real" [1]. Por más semejanzas que tenga, las máquinas virtuales no son computadoras ordinarias. Por esto se descubre que el rendimiento de una maquina virtual no es la misma que la de una computadora común. Este defecto es compensado por la capacidad de las máquinas virtuales de correr sobre cualquier equipo. Esto implica que una máquina virtual se puede cerrar (apagar), transportar en un dispositivo de almacenamiento portable a otro equipo y la máquina virtual que se abre es exactamente la misma.

El núcleo de una máquina virtual recibe el nombre de monitor de máquina virtual (MMV) o hipervisor, que corre sobre un sistema operativo y un hardware físico real sin interrumpir las operaciones normales del equipo anfitrión. Por esto, es posible ejecutar varias máquinas virtuales distintas, sobre una sola máquina real, con sistemas operativos diferentes. La mejoría que nos ofrece esta característica es que se pueden correr procesos, que requieren de SO distintos al mismo tiempo. De esta forma se pueden aprovechar las cualidades propias de cada sistema operativo, sin tener que cambiar de máquina [1].

Algunos se preguntarán ¿Por qué usar máquinas virtuales? Algunas razones son que ocasionan ahorros en espacio de memoria física, mejor aprovechamiento de los recursos disponibles y reduce costos de mantenimiento de equipo. Nos permite mejorar el aprovechamiento del equipo físico porque en general, nunca se llega a utilizar todos los recursos de una máquina, al mismo tiempo. Las máquinas virtuales nos dan la posibilidad de correr otras máquinas en la computadora física, utilizando los recursos que de otra forma estarían ociosas. Nos permite utilizar cualquier equipo disponible ya que el funcionamiento de la máquina virtual no depende de que todas las partes estén en un mismo chasis. Se pueden conectar máquinas con defectos para que estas se complementen y sean capaces de soportar la máquina virtual. Reduce costos de mantenimiento por razones obvias. Es mucho más económico contratar un técnico para revisar un solo equipo físico.

Además, podemos aludir la seguridad que nos proveen las máquinas virtuales. Cada máquina virtual es totalmente independiente de la computadora anfitrión. Esto significa que varias personas pueden utilizar la misma máquina física, y no están en riesgo de perder o exhibir información confidencial [9]. Lo anterior se debe a que cada máquina virtual se encuentra aislada de los demás SO presentes en el equipo.

También podemos mencionar servicios que brindan las máquinas virtuales a su SO anfitrión para depurar errores, probar aplicaciones nuevas en otro entorno, así como probar el funcionamiento de páginas Web en ambientes diferentes [1]. Ciertos patrones aprovechan al máximo sus servidores creando servidores sobre una máquina virtual. También unen todos los servidores de la empresa, para evitar el desperdicio que es tan común, y reduciendo los requerimientos de hardware en un 40% [2].

Las maquinas virtuales crean una computadora casi idéntica al la máquina sobre la cual esta montada. Una manera de ahorrar y aprovechar al máximo una máquina sería montando una sola computadora física con múltiples maquinas virtuales trabajando en diferentes tareas [9]. Cada maquina (aun estando dentro de la misma computadora) procesa sus trabajos sin obstruir la labor de los demás. Esto significa que se pueden elaborar diferentes proyectos y ejecutarlos dentro de una misma máquina, acercándose al aprovechamiento total del equipo físico.

La planificación de recursos aprovechados por dos (o más) sistemas operativos, corriendo sobre una máquina virtual, nos puede ayudar a ahorrar tiempo y espacio. Lo anterior se debe a que la utilización de máquinas virtuales implica comprar menos equipo físico ya que se corren varias maquinas virtuales en el misma máquina física. Esto siempre y cuando los picos no se den al mismo tiempo. Se debe tener cuidado a la hora de asignar tareas para no sobrecargar el hardware disponible. La proporción entre las múltiples máquinas virtuales queda a responsabilidad del usuario. Una solución para evitar la sobrecarga de la máquina es particionar el disco duro y correr cada máquina virtual en su propia partición. De esta forma cada una utiliza solamente los recursos disponibles en su partición lo cual facilita el trabajo de definir la asignación de recursos a cada máquina [9].

3.2 Nacimiento de las Máquinas Virtuales

El nacimiento de las máquinas virtuales se dio en MIT (Tecnological institute of Massachussets) con los estudios realizados sobre sistemas de tiempo compartido compatibles. Buscaban permitir que se ejecutaran varios sistemas operativos en un sólo hardware. Para esto separaron las siguientes funciones: multiprogramación y abstracción de hardware. El primer paso fue determinar los requisitos mínimos de hardware para soportar la virtualización [1].

La realidad del momento era uno en donde los equipos eran escasos y muy caros. En este momento tenia sentido crear varias máquinas sobre un sólo hardware ya que era difícil adquirir equipo. Con el auge de las PC, a un costo más bajo, su desarrollo quedo estancado. Luego, a principios de los 90 volvió la necesidad. Se hizo necesario bajar costos administrativos y disminuir dispersión de equipos de bajo costo. Al surgir necesidad, también surgió el hardware necesario para soportarlo. Tanto Intel como AMD lanzaron modelos para soportar tecnología virtual, es decir, separación lógica de los dispositivos físicos [1].

Así, la maquina virtual de Java nació gracias a la iniciativa de Sun Microsystems. Se creó un proyecto secreto llamado “The Green Project” integrado por 13 personas [8].

Se estudiaba la posibilidad de desarrollar un sistema de comunicación entre diferentes equipos. Lo que descubrieron fue que esto era casi imposible por las diferencias lógicas entre cada equipo, y la comunicación solo podría lograrse imponiendo una norma entre fabricantes para crear equipos y sistemas compatibles. La solución que encontraron a este problema fue la creación de una maquina virtual [8].

Esta maquina virtual seria portable y podría ejecutarse sobre cualquier dispositivo. Todas las maquinas virtuales comparten un código maquina: “bytecode”. De esta forma se logró la compatibilidad sin necesidad de modificar la arquitectura actual de los equipos [8].

El resultado de 18 meses de trabajo, aislado de Sun Microsystems fue un lenguaje llamado Oak, que luego paso a llamarse Java. Inicialmente se pensó que se le podría dar uso domestico, principalmente en televisores. Pero las compañías de cable no estaban preparadas para esta tecnología. Luego, con el auge de Internet y el lanzamiento de Netsacpe Navigator, implementado en java, se lanzó al mundo de desarrolladores de software [8].

Una de las facilidades que presenta java es la libertad de desarrollar en una plataforma y poder ejecutar en cualquier otra plataforma. Además permite la creación de aplicaciones web, programas muy personalizados y aplicaciones para artículos de uso frecuente. Incluso se usa en las aplicaciones usadas en los teléfonos móviles. Todo esto, gracias al uso de una máquina virtual. [4]

3.3 Interacción entre el Monitor de Maquina Virtual y el Sistema Operativo

Un MMV maneja los recursos físicos y provee la abstracción necesaria para uno o más máquinas virtuales. La Figura 1 muestra la estructura de una máquina virtual común. Como se ve en la figura, el SO anfitrión se comunica directamente con el hardware y las aplicaciones corren sobre el SO. La máquina virtual se encuentra encerrada por un círculo de líneas punteadas. Las aplicaciones del SO de la máquina virtual (o SO invitado) corren sobre el SO de la máquina virtual, el cual a su vez se corre sobre la MMV el cual provee la abstracción para la comunicación con el hardware. [19]

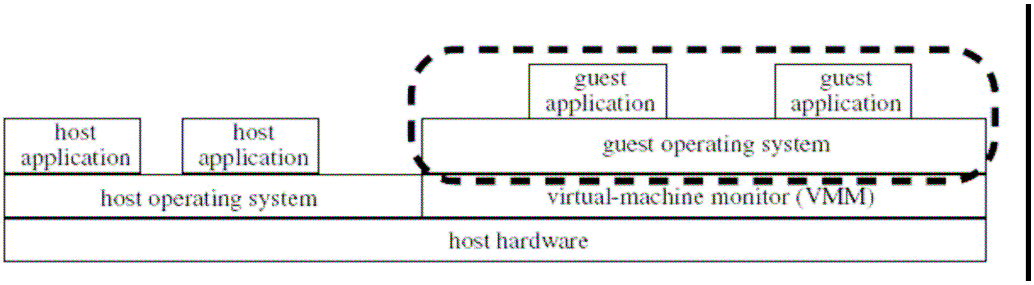


Figura 1: Estructura común de un MMV en la actualidad

Las instrucciones que da un sistema operativo al procesador operan en dos modos: el modo usuario, y el modo privilegiado (o modo kernel). Típicamente, en el procesador hay un bit de modo que indica en cuál modo se debe ejecutar cada instrucción [14]. El modo privilegiado tiene más acceso a los recursos de hardware de la máquina real; en cambio, el modo usuario tiene algunas restricciones de seguridad para evitar interferencia entre diferentes procesos que corren en la máquina. Entonces, ¿qué tiene que ver esto con máquinas virtuales?

En el procesador, las instrucciones que vienen del sistema operativo (como las llamadas al sistema) se ejecutan en modo privilegiado. En cambio, en una máquina virtual las instrucciones que usualmente se ejecutarían en modo privilegiado realmente se ejecutan en modo usuario porque aunque es un SO común y corriente el que está ejecutando tal código, para la máquina real es una aplicación más que está montada sobre el sistema operativo anfitrión. Dicho de otra manera, las instrucciones generadas por los sistemas operativos invitado (corriendo sobre máquinas virtuales) son vistas por el sistema operativo anfitrión como instrucciones de cualquier aplicación común, y son enviadas al procesador para ejecutarse en modo usuario.

Las máquinas virtuales funcionan en modo usuario porque son para el sistema operativo anfitrión iguales a cualquier otra aplicación. Esto implica que las máquinas virtuales acceden a los recursos de hardware del sistema como lo haría cualquier otra aplicación: a través de llamadas al sistema operativo anfitrión.

El MMV exporta abstracciones de hardware a la máquina virtual usando hardware emulado. La máquina virtual interactúa con el hardware virtual de la misma manera en que con hardware real. Estas interacciones son atrapadas por el MMV y emuladas en software. [19]

Un MMV soporta varios SO multiplexando el hardware de la computadora y proveyendo la ilusión de distintas máquinas virtuales cada una de las cuales corre un SO. El MMV aísla los recursos de cada máquina virtual por lo que puede “mapear” dos discos virtuales en diferentes sectores de un mismo disco físico. [19] Comparte los demás recursos del equipo físico de manera similar. [19]

Sin embargo, hay algunos dispositivos de hardware, como tarjetas de video y tarjetas de red, que a veces no son accesibles desde la máquina virtual sin un controlador adicional. Por ejemplo, la salida de video en el software de virtualización MobiDesk (descrito más adelante) está diseñada como un controlador de dispositivo de video virtual que intercepta los comandos de video en la capa de hardware del sistema anfitrión y provee un dispositivo de video separado por cada máquina virtual. En vez de mandar comandos de video al hardware local, el controlador de video virtual empaca comandos de video asociados con un usuario (máquina virtual) y los envía a la salida de video del cliente mediante una llamada al sistema operativo anfitrión. [15]

Hay casos en los que el desempeño de la máquina virtual puede ser mejorado mediante el uso de controladores adicionales [15], como es el caso del software de virtualización Xen, del que hablaremos más adelante.

Un software de virtualización debe ofrecer una abstracción de los recursos del sistema operativo anfitrión para poder brindar los servicios de computación necesarios para las aplicaciones corriendo en el SO invitado [16]. Esto se logra dando a cada sesión virtual (o máquina virtual) un dominio privado de recursos que permiten a los procesos de la sesión acceder los servicios del sistema operativo anfitrión. El dominio de cada sesión debe ser privado para poder brindar seguridad a los procesos que corren dentro de él, y para poder separar las abstracciones de datos de cada máquina virtual; esta privacidad significa

que un proceso no puede conocer o acceder ningún otro proceso de un dominio diferente al suyo, pero procesos dentro del mismo dominio sí pueden comunicarse mediante los tradicionales protocolos de comunicación entre procesos (*IPC- Inter-Process Comunication*).

3.4 Estructura típica de una máquina virtual

Para ilustrar la típica implementación de una máquina virtual de plataforma, tomaremos como ejemplo el software de virtualización Xen. Es frecuente el uso de la expresión MMV para detallar un software de virtualización como Xen o VMWare, dado que la aplicación principal trabaja como un monitor para todas las máquinas virtuales creadas sobre un computador.

Xen cuenta con un componente llamado hipervisor (MMV), el cual está en el nivel más bajo de la aplicación y tiene acceso directo al hardware. Sobre el MMV están montados los dominios (máquinas virtuales) ejecutando instancias de SO invitados. Cada SO invitado utiliza un segmento de memoria física preconfigurado (durante la creación de esa máquina virtual) [13]. Pero existe un dominio especial, llamado Dominio0, que compone una interfaz entre el MMV y el resto de dominios. Es el Dominio0 el que lleva a cabo las tareas de crear, terminar y migrar otras máquinas virtuales (Dominios Usuario o *DomU*). La figura1 presenta una ilustración de la estructura del monitor de máquinas virtuales Xen.

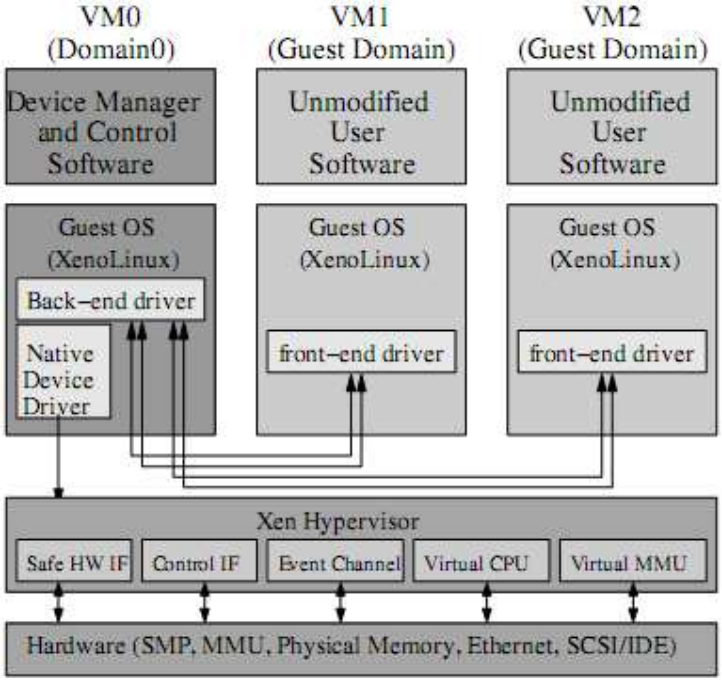


Figura2: Estructura del Monitor de Máquinas Virtuales Xen

La virtualización de dispositivos y entrada/salida (I/O) en Xen consta de un modelo de división de controladores [13]. El controlador de dispositivos nativo (el mismo del sistema operativo anfitrión) se ejecuta desde el Dominio0. El Dominio0 provee una interfaz con el controlador de dispositivos nativo (el mismo del sistema operativo anfitrión) llamado *back-end driver* que corre como un proceso y responde a las peticiones de acceso de cada DomU. El SO invitado en cada DomU utiliza un *frontend driver* para comunicarse con el backend.

La comunicación entre los dominios sucede a través de páginas de memoria compartidas y canales de eventos que proveen un mecanismo de notificaciones asíncronas [13]. Para enviar información a otro dominio, o para solicitar una operación de I/O, un dominio habilita el acceso remoto a sus páginas de memoria locales registrando las páginas en una tabla de accesos (llamada *grant table*) controlada por el MMV. Después de inspeccionar las páginas, se crea una referencia para cada página física, la cual es empleada por el segundo dominio para copiar las páginas a su espacio de memoria local.

Como ya se había mencionado las máquinas virtuales funcionan como cualquier otro proceso usuario, sin embargo, el código del hipervisor es ejecutado en modo kernel, el cual le da acceso completo al hardware, pero los dominios operan en modo usuario.

4. Virtualización de recursos

Ya se mencionaron algunas características de las máquinas virtuales, pero este no es el único uso que se le puede dar a la virtualización. Otra aplicación es la virtualización de recursos, lo cual consiste en aprovechar al máximo los recursos disponibles usando la virtualización para "crear" los recursos faltantes. Las Redes virtuales se basan en esto para crear redes complejas sin invertir en dispositivos de red. A continuación se expondrá el funcionamiento general de una red virtual.

4.1 Red Virtual

Redes virtuales o "Virtual private networks" (VPN) son redes privadas que usan una red publica (Internet) para conectar usuarios o oficinas lejanas a la red de la organización de forma segura. En lugar de usar conexiones reales, VPN usa conexiones virtuales direccionadas a través de Internet desde la red privada de una compañía, por ejemplo, hasta la red o sitio de un empleado. Tiene la funcionalidad de una red compleja, pero a menor precio. [7]

Esta tecnología tiene muchas aplicaciones, algunas de las cuales son: acceso remoto a computadoras por medio de Internet, participación interactiva de escritorio (en ingles: Desktop sharing, lo cual permite colaborar en tiempo real desde máquinas diferentes), presentaciones por Internet y permite compartir aplicaciones para probar software antes de bajar o comprar. [7] Una VPN debe soportar al menos las siguientes tres funciones: Acceso remoto por Internet de conexiones cliente, comunicación entre LAN y acceso controlado dentro de una intranet. [6]

"Desktop sharing" funciona enviando paquetes de información desde el computador anfitrión a la computadora remota describiendo que hay en la pantalla en cualquier momento dado. Los datos viajan por Internet, y sólo envía información que ha sido modificada, minimizando el ancho de banda necesario. [7]

Una VPN funciona usando infraestructura pública manteniendo privacidad. Protege la seguridad de la información usando un protocolo llamado "Layer two Tunneling Protocol" (L2TP). Funciona encriptando los datos y desencriptando al recibirlos a través de un "túnel" de datos por el que sólo pueden viajar datos debidamente encriptados. Además, para mayor seguridad, se pueden encriptar y desencriptar las direcciones de red. [6]

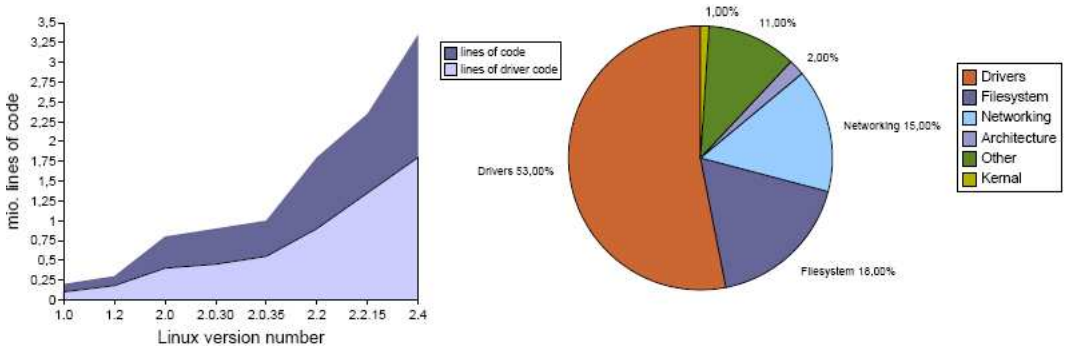
5. Drivers: problemática y soluciones

Un driver es un programa que le dice al sistema operativo como debe comunicarse con un nuevo periférico. Pero ¿Cómo manejan las máquinas virtuales a los driver?

Generalmente, estos se corren sobre la maquina virtual, y no sobre el MMV debido a que presentan una gran cantidad de errores. La única manera de proteger al MMV de estos errores es separándolo. La desventaja de esta separación es pérdida de rendimiento ya que el driver se encuentra en otro driver de maquina virtual, por lo tanto, otra dirección de memoria lo cual implica perdida de tiempo durante invocaciones. [16]

La alternativa es correr los drivers directamente sobre el MMV. Aunque esto resulten mejor rendimiento, sufre de perdida de garantía de seguridad e incurre en el costo de mantenimiento de soporta a infraestructura para el driver. Esta alternativa es la más usada. [16]

Además de la alta ocurrencia de errores en los drivers, estos son programas enormes, de gran cantidad de líneas de código. Asimismo, hardware real es muy complejo, lo que lleva a un código complejo. Por estas dos características los errores duran años en ser corregidos, y a veces no se logra de la mejor manera. [17] Podemos observar estos dos problemas en la Figura 3 que se muestra a continuación. Los problemas mencionados comprometen la seguridad del SO anfitrión.



a) Relación entre líneas de código del kernel Linux y su correspondiente código para el manejo de drive

b) Proporción de pulgas por tipo de código en Linux

Figura3: Problemas de drivers

Se proponen varias soluciones a esta problemática. La primera es reescribir los drivers, lo cual es un proceso largo y se expone a cometer nuevos errores. Además, es muy caro y no se logra una seguridad completa. El segundo es aislar completamente drivers en los que no se puede confiar, previniendo cualquier comunicación entre el driver y el exterior, esto es llamado Sandboxing. La tercera es encriptar todos los datos, logrando que el driver sea incapaz de ver cualquier información no encriptada, esto no soluciona los problemas y además requiere de que muchos drivers sean cambiados y también de nuevo hardware o bien extensiones de los protocolos. La cuarta solución es virtualizar el hardware. Esto quiere decir que los dispositivos de entrada y salida deben conocer las máquinas virtuales para permitir una comunicación directa sin pasar por el MMV (en donde se encuentran los drivers) y asignando mayor grado de inteligencia dentro del hardware. [17] En la figura numero 4 se muestra la relación entre la maquina virtual, el hardware, el MMV y los drivers para cada una de las soluciones propuestas.

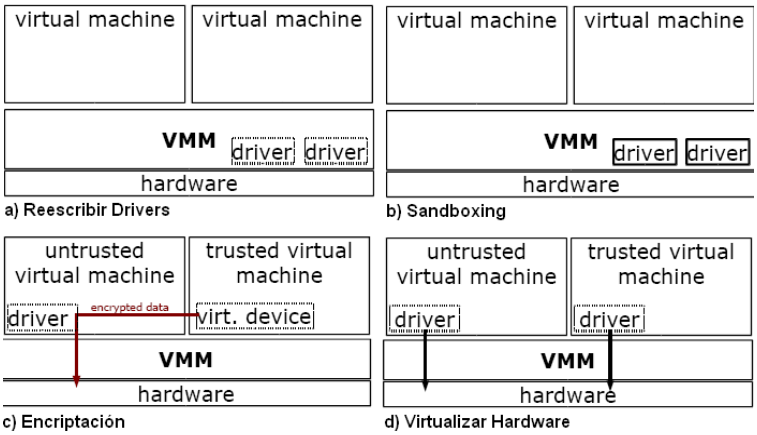


Figura4: Soluciones a los problemas de drivers en las MV

Las soluciones ofrecidas no son absolutas, es decir, nunca se puede confiar en un driver plenamente debido a su complejidad y tamaño. Además requieren que el hardware sea más inteligente.

5. Ejemplos de uso de la virtualización

A continuación se describirán ejemplos de las aplicaciones de virtualización mencionadas anteriormente. La máquina a describir fue una de las primeras de su tipo en ser lanzadas. Es de especial interés por estar basado en un lenguaje muy conocido por desarrolladores y estudiantes del área de la informática. Luego se describirá una aplicación llamada MobiDesk la cual implementa "Desktop sharing", para entender mejor el uso de una red virtual.

5.1 La Máquina Virtual de Java

Cuando se implementa un programa en C o C++ por ejemplo, esta sólo se puede utilizar sobre la plataforma de hardware sobre la cual fue desarrollada. Debido a que el código generado por el compilador en lenguaje de máquina es específico a esa plataforma [3]

A diferencia de la mayoría de los lenguajes de programación, Java utiliza un tipo especial de código de maquina llamado "bytecode", el cual utiliza un tipo especial de microprocesador [5]. En la época en la que se mercadeo por primera vez Java, no existía este microprocesador.

Debido a esto se decidió emular el microprocesador mediante una maquina virtual. La Máquina Virtual de Java (JVM por sus siglas en inglés: Java Virtual Machine) no corre sobre el microprocesador, sino que es emulada por el microprocesador de la computadora [5]. La JVM es la clave de la autonomía de los programas Java ya que se ejecutan independientes del SO y el hardware, además de que Java es un lenguaje poderoso el cual consta de un lenguaje de programación simple, orientado a objetos, con verificación estricta de tipos de datos, múltiples hilos, con ligado dinámico y con recolección automática de basura [3].

La JVM tiene la responsabilidad de interpretar el "bytecode" Java y trasformarlo a llamadas al sistema, así el desarrollador o programador no debe de preocuparse por el sistema operativo o el hardware del CPU ya que esto es irrelevante debido a que la JVM se encarga de la comunicación y sincronización de ellos.

La representación del código generado por el lenguaje Java (bytecode) es simbólico, lo cual significa que "los desplazamientos e índices dentro de los métodos no son constantes, sino que son cadenas de caracteres o nombres simbólicos" [3]. Así el nombre simbólico se busca en el .class correspondiente a la clase y se busca el adecuado valor numérico que determina el desplazamiento. Debido a esta metodología es posible introducir un nuevo método o sobrescribir uno existente en tiempo de ejecución [3].

Las especificación de la maquina virtual de Java define el formato de los archivos .class, así como la semántica de cada una de las instrucciones que comprenden el conjunto de instrucciones de la máquina virtual [3]. A esta implementación de la máquina virtual se le conoce como "Sistema en Tiempo de Ejecución Java" la cual incluye cosas como un motor de ejecución, administrador de seguridad, administrador de la memoria, administrador de errores y excepciones [3], entre otras cosas que no mencionaremos ya que no serán incluidos ni discutidos en este articulo.

Una descarga gratuita del controlador de java (incluyendo su máquina virtual) está disponible en la siguiente página: http://www.java.com/es/. También se ofrece una pequeña descripción y ayuda para los que no puedan descargar correctamente.

5.2 MobiDesk

MobiDesk es una PC (computadora personal) móvil virtual. Utiliza la red para separar la sesión del usuario del dispositivo moviendo toda lógica de aplicación a proveedores de "hosting". De esta manera el dispositivo sirve únicamente para enviar información nueva y mostrar información existente. También separa la sesión del sistema operativo usado permitiendo migrar toda la sesión de un servidor a otro. Esto es útil si se le tiene que dar mantenimiento a un servidor, el tiempo abajo se minimiza así como el impacto sobre los usuarios. Al terminar el mantenimiento, se vuelve a migrar al servidor, sin perder la conexión de los usuarios conectados. [15]

MobiDesk logra implementar los servicios antes mencionados introduciendo una delgada capa de virtualización entre el ambiente computacional del usuario y el sistema base. Enfoca en tres sistemas claves: despliegue, sistema operativo y red. Virtualiza recursos de despliegue proveyendo un driver que eficientemente codifica y redirecciona las actualizaciones del despliegue del servidor al dispositivo del usuario. Virtualiza los recursos del sistema operativo proveyendo un "namespace" virtual privado para cada sesión de usuario. Virtualiza la red proveyendo identificadores de direcciones virtuales para conexiones y un mecanismo independiente de trasporte para Proxy. [15]

Una descarga gratuita del controlador de MobiDesk para palm está disponible en la siguiente página: http://www.mrandersonmd.com/2006/10/09/software-infaltable-en-tu-palm-tx-i/. Además ofrece actualizaciones para este software.

6. Amenazas de la Virtualización

Los siguientes puntos son considerados amenazas porque comprometen la seguridad de los sistemas involucrados. Pero esta tecnología sigue evolucionando, y algunas de estas amenazas podrían convertirse en beneficios.

6.1 Comunicación entre máquinas virtuales o entre máquinas virtuales y el anfitrión

Por los diferentes usos que se le pueden dar a las maquinas virtuales, es de vital importancia tener un control de si la maquina virtual que se esta utilizando permite tener acceso a la computadora anfitriona o a otras maquinas virtuales dentro del mismo anfitrión. [18]

Existen tecnologías en donde se puede compartir un "clipboard" para permitir transferir información entre la maquina virtual y el anfitrión. Esta tecnología, además de ser muy ventajosa, podría permitir que programas malignos transmitan información a través de este "clipboard" al utilizarlo como un tipo de puerta de entrada, enviando información desde o para el SO del anfitrión. [18]

También, existe una tecnología de maquina virtual que permite al kernel del SO enviar información sobre entradas del teclado y actualizaciones de la pantalla a través de terminales virtuales hacia otra maquina virtual. Esta información se almacena en el anfitrión y permite que se puedan monitorear los movimientos de las diferentes maquinas virtuales. [18]

Hay otras tecnologías de maquinas virtuales que ni siquiera contemplan el aislar a unas maquinas virtuales de otras o al anfitrión. [18]

6.2 Escape de la máquina virtual

Las máquinas virtuales permiten compartir recursos computacionales, siempre aislando la máquina virtual de su anfitrión. Idealmente, la máquina virtual no debe poder ver, monitorear ni afectar al anfitrión, pero debido a limitaciones de arquitectura, tipo de aislamiento implementado por el vendedor o errores en el software esto no siempre se cumple. [18]

El peor caso es que un programa corriendo dentro de la máquina virtual sobrepase la capa de la máquina virtual y logre tener acceso total del anfitrión. De aquí en termino "escape de la maquina virtual". [18]

6.3 Monitoreo de máquina virtual desde el anfitrión

No se considera una limitación el poder monitorear una máquina virtual desde el anfitrión, pero esto implica que el anfitrión necesita seguridad más estricta que las máquinas virtuales. El anfitrión es capaz de controlar desde cuando se inicia y apaga hasta el monitoreo de aplicaciones y recursos disponibles para la máquina virtual. Además de copiar, consultar y hasta modificar la información en el disco de la máquina virtual. Todo paquete de red direccionado a la máquina virtual pasa por el anfitrión, por lo que también monitorea el tráfico de red. [18]

6.4 Monitoreo de máquina virtual desde otra máquina virtual

Debido a la gran importancia que tiene el aislar a las maquinas virtuales entre ellas en un mismo anfitrión, se considera un fallo si se permite que de alguna forma una maquina virtual tenga acceso a recursos que le pertenecen a otra maquina virtual o bien al mismo anfitrión (de no ser configurada explícitamente para ello). El hipervisor es responsable de aislar la memoria entre las maquinas virtuales. [18]

Cuando se implementan redes virtuales y una maquina virtual utiliza un hub virtual o bien un switch virtual para conectar a todas las maquinas virtuales con el anfitrión, cabe la posibilidad de que la

maquina virtual que utilizo el hub/swtich pueda filtrar o bien leer paquetes de otras maquinas virtuales en el proceso de transmisión de los paquetes. [18]

Se puede utilizar autentificación para reducir el daño, también se pueden establecer limites sobre la dirección MAC que se usa en cada interfaz de la red virtual. [18]

6.5 Negación de servicios

Existe el problema de que las maquinas virtuales en un mismo anfitrión comparten los recursos de hardware entre ellas mismas y el anfitrión. Esto puede causar que una de las maquinas virtuales (o bien el anfitrión) absorba una gran cantidad de los recursos causando que se de un ataque de negación de servicio hacia otra maquina virtual. [18]

Para resolver este problema se pueden asignar recursos fijos a cada maquina virtual, o bien establecer un limite de la capacidad de recursos a lo que una maquina virtual puede acceder con el fin de evitar que las otras maquinas virtuales se queden sin recursos. [18]

7. Conclusión

La virtualización permite un mejor aprovechamiento de recursos, es más eficiente y brinda mayor seguridad a menor costo.

Las máquinas virtuales sirven tanto para aprovechar los recursos de un equipo (corriendo los procesos de dos ordenadores lógicos, con capacidades diferentes), como para ampliar la producción de una red de máquinas físicas (Por ejemplo, en una red de servidores, con espacio no usado se puede establecer una maquina lógica para acelerar el trabajo) rindiendo al máximo los recursos disponibles.

Las aplicaciones de las máquinas y redes virtuales son diversas y muy comunes, aunque en general no se conoce este dato, como es el caso de la máquina virtual de java. Muchas personas la utilizan sin saber porqué es tan portable.

Los drivers son esenciales para el funcionamiento de las máquinas virtuales, pero también presentan uno de los mayores peligros de los mismos. Las soluciones existentes no presentan un arreglo completo de los drivers, sólo se puede dar un uso cuidadoso para evitar la problemática mencionada.

Las ventajas que nos presenta la virtualización son amplias, pero como todo tiene limitaciones y amenazas. Siempre que se le de el manejo correcto, se logrará obtener el mayor beneficio con pérdidas mínimas.

8. Bibliografía

[1] Zorraquino, F.J. Virtualización: Maquina Virtual. *Astic*, URL, 2006, pp. 68 – 77

[2] Ramos, S. Máquinas Virtuales: Virtualización del Hardware. *Jeuazzaru*, URL: http://www.jeuazarru.com/docs/Maquina_virtual.pdf, 6 de Diciembre, 2005, pp. 1 - 13

[3] Menchac, R. Arquitectura de la Máquina Virtual de Java. *Revista Digital Universitaria*, URL: http://www.revista.unam.mx/vol.1/num2/art4/, Vol.1, No.21, de Octubre, 2000

[4] Sun Microsystems, Acerca de la tecnología de java. *Java*, URL: http://www.java.com/es/about/

[5] Reilly, D. Inside java: The java Virtual Machine. *Java Coffee Break*, URL: http://www.javacoffeebreak.com/articles/inside_java/insidejava-jan99.html, 05 de Junio, 2006

[6] http://searchsecurity.techtarget.com/sDefinition/0,,sid14_gci213324,00.html

[7] http://communication.howstuffworks.com/how-desktop-sharing-works.htm

[8] BarraPunto, Introducción Breve Historia de Java, *Entorno de acs,* URL: https://acs.barrapunto.org/svn/pfc/EntornoJava/Java.pdf, 26 de Mayo, 2007, pp. 1 – 29, http://www.ac.upc.edu/pub/reports/DAC/2003/UPC-DAC-2003-5.pdf

[9] Meier, W. Linux en el amanecer de una era virtual: Todo en uno. Linux – Magazine. https://www.linux-magazine.es/issue/23/Virtualizacion.pdf, pp. 13 - 15

[10] Pizzonia, Mauricio. Netkit: Easy emulation of complex networks on inexpensive hardware. Ddept. Of computer science and automation, Roma tre University.

[11] http://es.wikipedia.org/wiki/Virtualizaci%C3%B3n

[12] http://www.vmware.com/company/

[13] Virtual Machine Aware Communication Libraries for High Performance Computing, Wei Huang Matthew J. Koop Qi Gao Dhabaleswar K. Panda, Network-Based Computing Laboratory, The Ohio State University

[14] Silberschatz, A., Galvin, P. & Gagne, G. Sistemas Operativos, Sexta Edición, Editorial Limusa, México, 2002.

[15] Ricardo A. Baratto, Shaya Potter, Gong Su, Jason Nieh: MobiDesk - Mobile Virtual Desktop Computing. Department of Computer Science, Columbia University, New York, NY, USA.

[16] omitted for blind review. Twin drivers: Automatic derivation of fast and safe hypervisor drivers from guest OS drivers. 14 de Agosto del 2008.

[17] Shcwarz, Felix. Virtual Machine monitors for the trusted computing base. *kbs.cs.tu-berlin.de/teaching/sose2005/tcb/folien/Virtual%20Machine%20**Monitors**.pdf -*

[18] Kirch, Joel. Virtual Machine security guidelines, The Center for internet security. September 2007

[19] King, Samuel. SubVirt: Implementing malware with virtual machines. University of Michigan.