

Sistemas Operativos Mínimos para la Nube

Barreiro Valdez Alejandro

21 de abril de 2022

1. Conceptos básicos

Algunos conceptos básicos para entender la utilidad y la aplicabilidad de los sistemas operativos mínimos para la nube tienen que ver con la virtualización. En esta sección se explican los conceptos básicos que tienen que ver con virtualización como qué es una máquina virtual, sus ventajas y desventajas, qué es un *Hypervisor* o hipervisor, los tipos de hipervisores que existen y qué es la nube o *cloud*. Con estos conceptos básicos se busca tener antecedentes para entender lo que es OS^V y su funcionalidad.

1.1. Virtualización

Tecnología que permite imitar las características del *hardware* y crear un sistema informático virtual. Esto permite ejecutar varios sistemas virtuales en un solo servidor. Tiene la función de simular la existencia de un recurso tecnológico que se quiere virtualizar. Existe la virtualización de plataforma, de datos, de recursos de hardware, de red, de entornos de software de ejecución y de escritorios entre otras. (Wikipedia, 2022b)



Figura 1: Virtualización

1.2. Máquina virtual

Es un ambiente virtual que trabaja de manera similar a una computadora dentro de otra. Corre en una partición aislada de la máquina *host* y tiene sus propios recursos virtuales. Los usuarios pueden correr aplicaciones en estas máquinas virtuales y usarlas como una estación de trabajo normal. Las máquinas virtuales son posibles gracias a las tecnologías de virtualización. Utilizando las técnicas de virtualización se pueden generar máquinas virtuales que corren en una sola máquina. La máquina física se denomina *host* y máquinas virtuales que corren en ella se denominan *guests*. (Citrix, 2022)

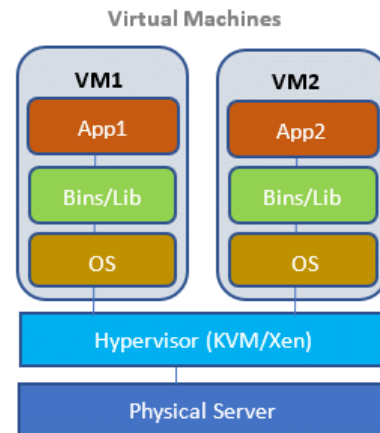


Figura 2: Máquina Virtual

1.2.1. Ventajas

- Se pueden crear múltiples sistemas operativos en el mismo servidor.
- Si la máquina virtual se afecta o crashea de alguna manera, esto no afecta a la máquina

host.

- Se pueden utilizar para cuestiones de seguridad. Si se va a correr una aplicación dudosa se puede correr en una máquina virtual y si esta causa algún daño esto no afectará a la máquina *host*.

1.2.2. Desventajas

- Son menos eficientes porque tienen que acceder al *hardware* de manera indirecta. El hipervisor tendrá que pedir recursos a la máquina *host* y esto genera pérdida de tiempo en las operaciones.
- En ciertas organizaciones se tienen muchas máquinas virtuales lo cual genera pérdida de recursos. Se debe tener a alguien capaz de saber cuáles realmente se necesitan y administrar las máquinas virtuales a utilizar.
- Si la máquina *host* tiene problemas o *bugs* esto puede afectar el funcionamiento de las máquinas virtuales. Si alguna de las máquinas virtuales está conectada con otro estos errores se pueden propagar.

(Totounji, 2022)

1.3. Hypervisor

Este software conocido como hipervisor maneja las máquinas virtuales y la virtualización que se puede realizar. El hipervisor es el responsable de manejar los recursos como memoria y almacenamiento de los *hosts* hacia los *guests*. También maneja los recursos de cada una de las máquinas virtuales y programa las operaciones para que no invadan los recursos de otras máquinas virtuales. Las máquinas virtuales solo sirven si existe un hipervisor que realice la virtualización y que distribuya los recursos del *host*.

1.3.1. Bare Metal Hypervisors

Se instalan de manera nativa en el *hardware* de la máquina. Las máquinas virtuales interactúan directamente con los *hosts* para asignar recursos sin capas adicionales de *software* entre ellos.

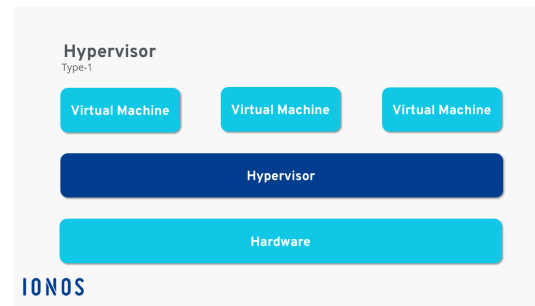


Figura 3: Hipervisor

1.3.2. Hosted Hypervisors

Estos hipervisores corren en el sistema operativo del *host*. Se le pasan los requisitos de la máquina virtual al sistema operativo *host* y se le proveen los recursos físicos necesarios a cada uno de los *guests*. Estos hipervisores son más lentos porque todas sus acciones tienen que pasar por el sistema operativo *host* antes.

1.4. Cloud

Tecnología que permite acceso remoto a *software*, procesamiento de datos y almacenamiento por medio de Internet. Esto genera una alternativa a no tener que ejecutar dicho *software* en un servidor local o computadora personal. Para este modelo, no se necesita instalar las aplicaciones que se usarán, están disponibles para su uso. (Force, 2022)



Figura 4: Cómputo en la nube

Las máquinas virtuales que corren en la nube utilizan los mismos sistemas operativos que se utilizan en máquinas físicas. Esta elección de sistemas operativos parece tener menos sentido porque estos sistemas operativos tienen soporte para una larga selección de *hardware*. Los sistemas operativos de las máquinas virtuales en la nube deberían ser rápidos, pequeños y fáciles de administrar a gran escala.

1.5. Java Virtual Machine (JVM)

Máquina virtual que le permite a una computadora utilizar programas de Java así como programas escritos en otros lenguajes que se compilan utilizando el *bytecode* de Java. La máquina virtual está especificada con una descripción que habla formalmente de lo que necesita esta máquina virtual. (Wikipedia, 2022a)

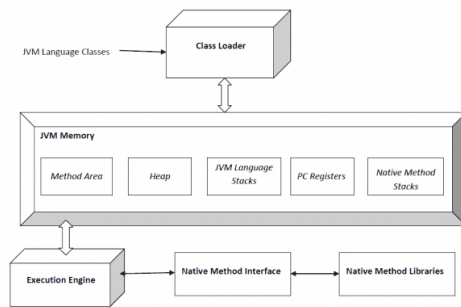


Figura 5: Especificaciones JVM

2. OS^V

Es un sistema operativo en la nube creado por las mentes detrás de KVM y su función es crear una pila de *software* creada para *hostear* una sola aplicación eliminando muchos niveles de abstracción.

2.1. ¿Qué es?

Este proyecto se define a sí mismo en su página de GitHub como un unikernel diseñado para correr aplicaciones solas y sin modificar de Linux de manera segura como una micro máquina

virtual utilizando un hipervisor. Está diseñado para correr binarios de x86-64 y AArch64 Linux haciéndolo un unikernel compatible con binarios de Linux. Junta una aplicación, un servidor de aplicación y una JVM en un hipervisor que conecta directo con el *hardware* sin la necesidad de la intervención de Linux. Su propósito es eliminar características duplicadas que existen en los sistemas operativos y los hipervisores de cloud. Su uso específico es correr una sola aplicación en una máquina virtual en la nube.

2.2. Compatibilidad con lenguajes

OS^V se utiliza para correr los sistemas en tiempo de ejecución de lenguajes como JVM, Python, Node.JS, Ruby, Erlang y aplicaciones que fueron construidas en esos tiempos de ejecución. También puede correr aplicaciones escritas en lenguajes como C, C++, Golang y Rust.

2.3. Hypervisors donde corre

Para construir y correr imágenes con un comando, los creadores recomiendan usar Caps-tan. Esta herramienta de construcción se pueden crear imágenes de máquinas virtuales que pueden ser corridas dentro de un ambiente de nube. Los hipervisores donde corre son los siguientes: QEMU/KVM, Firecracker, Cloud Hypervisor, Xen, VMWare, VirtualBox y Hyperkit. También se puede correr en *open clouds* como AWS EC2, GCE y OpenStack.

3. Preguntas Relevantes

3.1. ¿Qué gana OS^v y otros proyectos por el estilo? ¿Qué pierden?

En la presentación sobre este proyecto se habla de cómo el modelo actual sobre las capas en la nube es muy repetitivo. Existen tres capas diferentes que tienen la misma función de proveer protección y abstracción. Una ventaja que tiene

OSv es que liga todas estas capas redundantes con la filosofía de que menos es más. También, se utiliza solo un espacio de direccionamiento que le permite al JVM acceder directamente a las tablas de páginas. También, como solo se utiliza para una aplicación, se evitan algoritmos bloqueantes. Este sistema operativo reduce los gastos generales de memoria y el CPU que se generan al correr un sistema operativo tradicional en una máquina virtual. Los creadores comparan esta herramienta con los contenedores, pero dicen que obtienes la seguridad que solo se puede tener con virtualización verdadera. El tiempo de *boot* de la máquina virtual es muy poco, tiene un tamaño de imagen muy reducida y hay más memoria disponible para la aplicación. (Kivity, 2014)

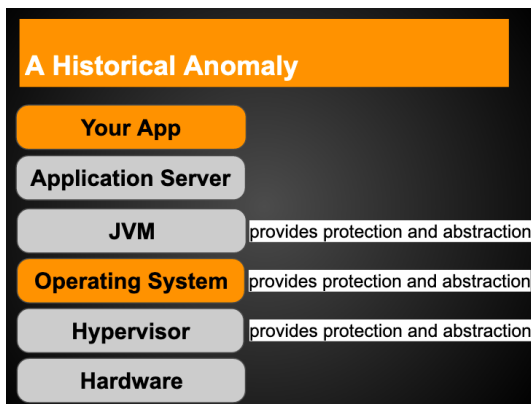


Figura 6: Capas redundantes

Se pierde el poder crear máquinas virtuales para sistemas operativos ya que la funcionalidad de este proyecto es utilizarlo para aplicaciones. Por lo tanto, todas las configuraciones que se desean realizar en un sistema operativo no se pueden realizar ya que se tiene uno que es minimalista. Esto puede ser visto como ventaja o desventaja, ya que sus creadores lo ven como ventaja porque esto ahorra tiempo. Además, al ser un programa en desarrollo cuenta con un soporte de *hardware* limitado. (Clark, 2013)

3.2. ¿Cómo ha avanzado la adopción de estas ideas?

Este proyecto sigue buscando a desarrolladores a nivel de kernel, que puedan administrar *stack* y con otras especificaciones en GitHub. Se considera que OSv tiene mejor rendimiento que algunas aplicaciones en Linux como SpecJVM, MemCacheD, Cassandra y TCP/IP. En varios campos se tiene un avance significativo pero se sigue desarrollando y vendiendo esta idea y proyecto. Se planea continuar explorando interfaces que añadir a OSv para mejorar el rendimiento de las aplicaciones. En vez de modificar aplicaciones individuales se modificarán ambientes como la JVM donde las aplicaciones corren para mejorar OSv. Este sistema operativo fue utilizado en MIKELANGELO, un proyecto que tenía el objetivo de cerrar la brecha entre la nube y HPC (*High Performance Computing*).

3.3. ¿Qué le agrega o quita a la complejidad de la administración de sistemas?

Este proyecto le quita diferentes capas de abstracción a las aplicaciones en la nube por lo que la administración de sistemas se vuelve más compleja ya que no se tienen abstracciones. Lo que se simplifica es la configuración del sistema operativo. No existen opciones para configurar el sistema operativo y personalizarlo por lo que ya no se pierde tiempo en esto.

3.4. ¿Qué significa esto para nuestra carrera/futuro profesional? ¿Hay de nuevo espacio para desarrollar sistemas operativos novedosos, o serán unos pocos los que llenarán el nicho?

Este tipo de desarrollos significan que se pueden seguir desarrollando sistemas operativos novedosos para el uso de otras tecnologías. En la

actualidad, existen mejoras en los sistemas operativos de uso común, pero las ideas novedosas se encuentran en desarrollar sistemas operativos con diferentes fines para que complementen otras tecnologías existentes.

Referencias

- Citrix. (2022). <https://www.citrix.com/es-mx/solutions/vdi-and-daas/what-is-a-virtual-machine.html>. *Citrix Solutions*. Descargado de <https://www.citrix.com/es-mx/solutions/vdi-and-daas/what-is-a-virtual-machine.html>
- Clark, J. (2013, 17 de septiembre). Kvm kings unveil 'cloud operating system'. *The Register*. Descargado de https://www.theregister.com/2013/09/17/cloudius_systems_osv_cloud_software/
- Force, S. (2022). Cloud computing: Aplicaciones en un solo lugar. *Cloud Computing*. Descargado de <https://www.salesforce.com/mx/cloud-computing/>
- Kivity, A. (2014, 19 de junio). Optimizing the operating system for virtual machines. *USENIX*. Descargado de <https://www.usenix.org/system/files/conference/atc14/atc14-paper-kivity.pdf>
- Totounji, A. (2022, 23 de febrero). Virtual machines: A closer look at pros and cons. *CYNEXLINK*. Descargado de <https://cynexlink.com/latest-articles/virtual-machines-pros-cons/>
- Wikipedia. (2022a). Java virtual machine. *Wikipedia.org*. Descargado de https://en.wikipedia.org/wiki/Java_virtual_machine
- Wikipedia. (2022b). Virtualización. *Wikipedia.org*. Descargado de <https://es.wikipedia.org/wiki/Virtualizaci%C3%B3n>