Introduction

Overview of Virtualization

Virtualization allows you to create multiple Virtual Machines (VMs) to run in isolation, side by side on the same physical machine.

Each virtual machine has its own set of virtual hardware (RAM, CPU, NIC) upon which an operating system and fully configured applications are loaded. The operating system sees a consistent, normalized set of hardware regardless of the actual physical hardware components.

In a virtual machine, both hardware and software are encapsulated in a single file for rapid provisioning and moving between physical servers. You can move a virtual machine, within seconds, from one physical server to another for zero-downtime maintenance and continuous workload consolidation.

The virtual hardware makes it possible for many servers, each running in an independent virtual machine, to run on a single physical server. The advantages of virtualization include better use of computing resources, greater server density, and seamless server migration.

Overview of Cisco Virtual Machine Fabric Extender

A virtualized server implementation consists of one or more VMs that run as guests on a single physical server. The guest VMs are hosted and managed by a software layer called the hypervisor or virtual machine manager (VMM). Typically, the hypervisor presents a virtual network interface to each VM and performs Layer 2 switching of traffic from a VM to other local VMs or to another interface to the external network.

Working with a Cisco virtual interface card (VIC) adapter, the Cisco Virtual Machine Fabric Extender (VM-FEX) bypasses software-based switching of VM traffic by the hypervisor for external hardware-based switching in the fabric interconnect. This method reduces the load on the server CPU, provides faster switching, and enables you to apply a rich set of network management features to local and remote traffic.

VM-FEX extends the IEEE 802.1Qbh port extender architecture to the VMs by providing each VM interface with a virtual Peripheral Component Interconnect Express (PCIe) device and a virtual port on a switch. This solution allows precise rate limiting and quality of service (QoS) guarantees on the VM interface.

Virtualization with a Virtual Interface Card Adapter

A Cisco VIC adapter is a converged network adapter (CNA) that is designed for both bare metal and VM-based deployments. The VIC adapter supports static or dynamic virtualized interfaces, which includes up to 128 virtual network interface cards (vNICs).

There are two types of vNICs used with the VIC adapter—static and dynamic. A static vNIC is a device that is visible to the OS or hypervisor. Dynamic vNICs are used for VM-FEX by which a VM is connected to a veth port on the Fabric Interconnect.

VIC adapters support VM-FEX to provide hardware-based switching of traffic to and from virtual machine interfaces.

Single Root I/O Virtualization

Single Root I/O Virtualization (SR-IOV) allows multiple VMs running a variety of guest operating systems to share a single PCIe network adapter within a host server. SR-IOV allows a VM to move data directly to and from the network adapter, bypassing the hypervisor for increased network throughput and lower server CPU burden. Recent x86 server processors include chipset enhancements, such as Intel VT-x technology, that facilitate direct memory transfers and other operations required by SR-IOV.

The SR-IOV specification defines two device types:

- Physical Function (PF)—Essentially a static vNIC, a PF is a full PCIe device that includes SR-IOV capabilities. PFs are discovered, managed, and configured as normal PCIe devices. A single PF can provide management and configuration for a set of virtual functions (VFs).
- Virtual Function (VF)—Similar to a dynamic vNIC, a VF is a full or lightweight virtual PCIe device that provides at least the necessary resources for data movements. A VF is not managed directly but is derived from and managed through a PF. One or more VFs can be assigned to a VM.

SR-IOV is defined and maintained by the Peripheral Component Interconnect Special Interest Group (PCI-SIG), an industry organization that is chartered to develop and manage the PCI standard. For more information about SR-IOV, see the following URL:

http://www.intel.com/content/www/us/en/pci-express/pci-sig-sr-iov-primer-sr-iov-technology-paper.html

Hypervisors that support SR-IOV include Linux KVM and Microsoft Hyper-V.

The following Cisco Virtual Interface Cards support SR-IOV with VM-FEX:

- Cisco UCS M81KR Virtual Interface Card
- Cisco UCS P81E Virtual Interface Card
- Cisco UCS Virtual Interface Card 1280
- Cisco UCS Virtual Interface Card 1240
- Cisco UCS Virtual Interface Card 1225
- Cisco UCS Virtual Interface Card 1225T

- Cisco UCS Virtual Interface Card 1227
- Cisco UCS Virtual Interface Card 1227T
- Cisco UCS Virtual Interface Card 1340
- Cisco UCS Virtual Interface Card 1380
- Cisco UCS Virtual Interface Card 1385

VM-FEX for KVM

Overview of VM-FEX for KVM

The Kernel-based Virtual Machine (KVM) is a virtualization package for Linux on an x86 hardware platform. KVM uses x86 hardware virtualization extensions (for example, Intel VT-x) to implement a hypervisor that hosts VMs as userspace processes. Cisco UCS servers support the KVM-based Red Hat Enterprise Virtualization (RHEV) as the hypervisor in a server virtualization system.

With VM-FEX for KVM, the RHEV hypervisor performs no switching of VM traffic. Working with an installed VIC adapter, the hypervisor acts as an interface virtualizer and performs the following functions:

- For traffic going from a VM to the VIC, the interface virtualizer identifies the source vNIC so that the VIC can explicitly tag each packet that is generated by that vNIC.
- For traffic that is received from the VIC, the interface virtualizer directs the packet to the specified vNIC.

All switching is performed by the external fabric interconnect, which can switch not only between physical ports, but also between virtual interfaces (VIFs) that correspond to the vNICs on the VMs.

For more information about KVM, see the following URL: http://www.linux-kvm.org.

Cisco UCS Manager Components

Cluster

The Cisco UCS cluster is a grouping of hypervisors that can be distributed across multiple hosts. In a KVM system, the cluster is analogous to the distributed virtual switch (DVS) in a VMware ESX system.

In the current Cisco UCS KVM implementation, the cluster defines the scope of the port profile and is the boundary of the migration domain. When multiple KVM hosts are associated to a cluster, you can migrate a VM from one host to another within the cluster.

Note In the current Cisco UCS implementation of VM-FEX for KVM, only one cluster, the default cluster, is used. Although you can create additional clusters, you can specify only the default cluster for a VM on the KVM host.

Port Profiles

Port profiles contain the properties and settings that are used to configure virtual interfaces in Cisco UCS. The port profiles are created and administered in Cisco UCS Manager.

Important After a port profile is created, assigned to, and actively used by a cluster, any changes made to the networking properties of the port profile in Cisco UCS Manager are immediately applied to the cluster with no need for a host reboot.

Port Profile Client

The port profile client is a cluster to which a port profile is applied.

Note In the current Cisco UCS implementation of VM-FEX for KVM, the default cluster is the only available port profile client.

KVM Components

Hypervisor

The hypervisor supports multiple VMs that run a variety of guest operating systems by providing connectivity between the VMs and the network. The hypervisor for KVM is a host server with Red Hat Enterprise Linux (RHEL) installed. The earliest supported release for VM-FEX is RHEL 6.1, but some features (such as SR-IOV) require a later version.

The hypervisor must have a Cisco VIC adapter installed.

For more information about virtualization using Red Hat Enterprise Linux, see the *Red Hat Enterprise Virtualization for Servers Installation Guide* available at the following URL: http://www.redhat.com.

libvirt

Libvirt is an open source toolkit that allows you to manage various virtualization technologies such as KVM, Xen, and VMware ESX. Libvirt, which runs on the hypervisor as a service named libvirtd, provides a command-line interface (virsh) and provides the toolkit for a graphical user interface package (virt-manager).

Each virtual machine created and managed by libvirt is represented in the form of a domain XML file.

For more information about the libvirt virtualization API, see the following URL: http://www.libvirt.org.

For more information about the virsh CLI, see the following URLs:

- http://linux.die.net/man/1/virsh
- http://www.libvirt.org/virshcmdref.html

MacVTap

MacVTap is a Linux driver that allows the direct attachment of a VM's vNIC to a physical NIC on the host server.

For more information about the MacVTap driver, see the following URL: http://virt.kernelnewbies.org/MacVTap.

VirtIO

The VirtIO paravirtualized network driver (virtio-net) runs in the guest operating system of the VM and provides a virtualization-aware emulated network interface to the VM.

For more information about the VirtIO driver, see the following URL: http://wiki.libvirt.org/page/Virtio.

Driver Topologies

Several driver topologies (modes) are available to implement a VM-FEX connection between a VM vNIC and the host VIC adapter. In each of these topologies, VM traffic is sent only to or from the VIC adapter. Traffic from one VM to another VM on the same host must first exit the host for switching by the external fabric interconnect.

Note In any topology, the configuration of the Quick EMUlator (QEMU) PCI layer might limit the number of PCI devices that the host can assign to a VM.

MacVTap Direct (Private)

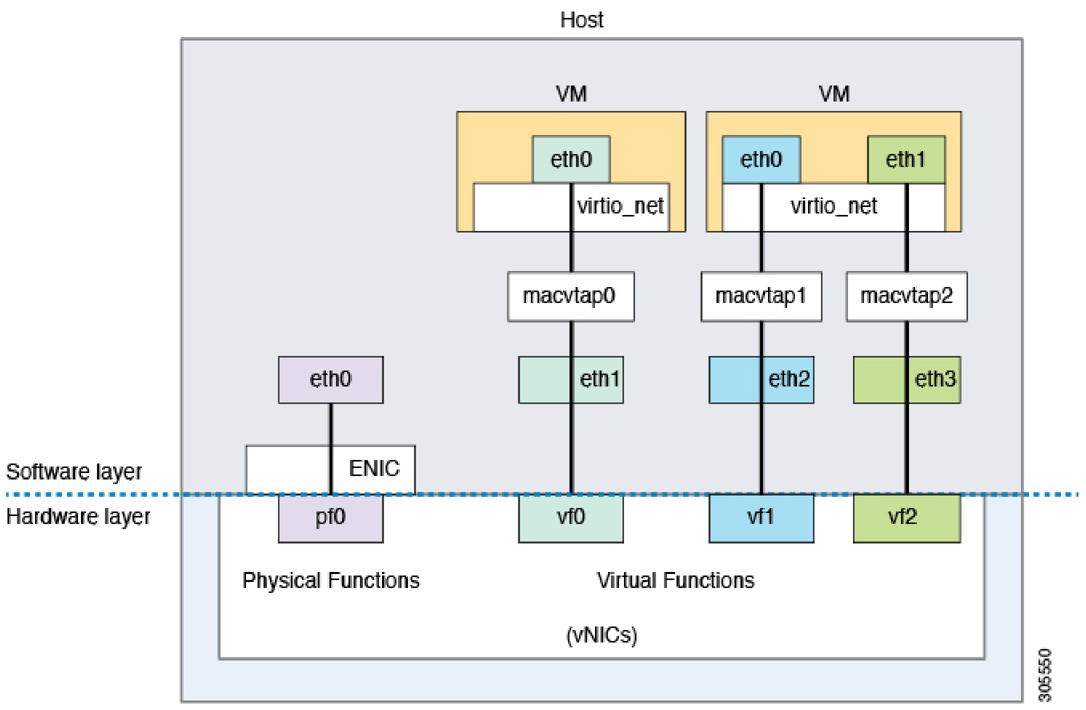
The MacVTap Linux driver is installed in the hypervisor (VMM) and connects each VM's VirtIO interface to a physical PCIe port of the VIC adapter. The MacVTap driver mode is private, which means that all VM traffic is sent directly to and from the host adapter with external switching. The number of supported VMs is limited to the number of VIC adapter ports. Live migration is supported.

Note Beginning with Cisco UCS Release 2.1, the MacVTap Direct (Private) topology is no longer supported.

SR-IOV with MacVTap Passthrough (Emulation Mode)

The MacVTap Linux driver is installed in the hypervisor and connects each VM's VirtIO interface to a VF on an SR-IOV-capable VIC adapter. The MacVTap driver mode is 'passthrough' and all VM traffic is sent to and from the VF. When we apply a port profile to a VF, libvirt determines the PF associated with the VF, and it configures the VF going through the PF. This topology is also known as MacVTap passthrough (emulation mode). An example of SR-IOV with MacVTap passthrough is shown in Figure 1. Figure 1 is a simplified version of the hardware and software components.

Figure 1: Figure 1



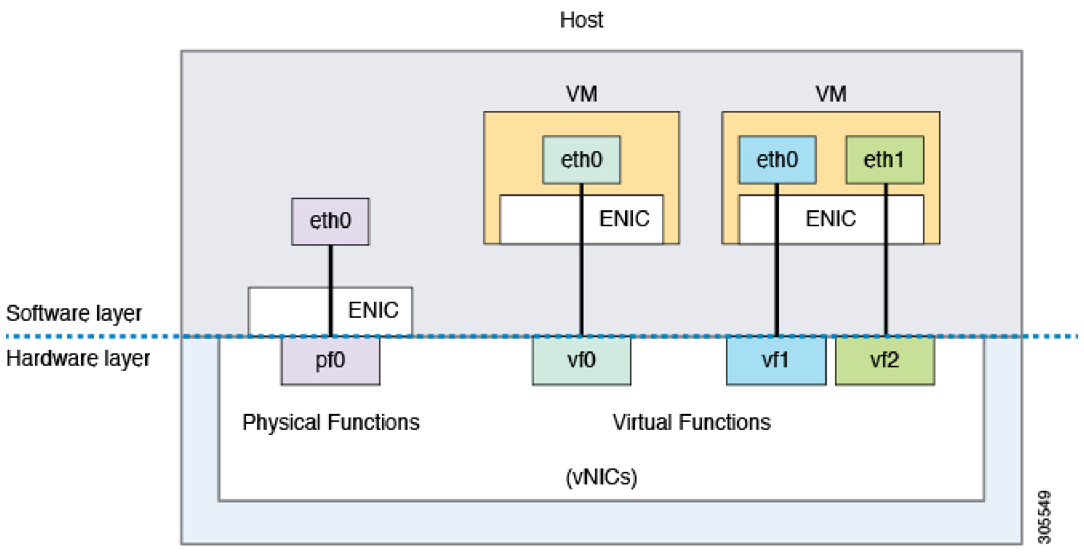
The maximum number of supported VMs is determined by the number of VFs provided by the VIC adapter. The number of VFs that you can assign to a PF might be further limited by the host Netlink protocol implementation (the limit is typically between 22 and 32 VFs per PF, depending on the OS version). Live migration is supported.

SR-IOV VF Passthrough (Hostdev Mode)

The MacVTap and VirtIO drivers are not used. Instead, the Ethernet driver (enic) of the VIC adapter is installed in the VM kernel and connects directly to a VF. You can configure the VF through the associated PF using libvirt. In libvirt documentation, this topology is called hostdev mode. This topology is also known as PCI passthrough. The number of supported VMs is determined by the number of VFs provided by the VIC adapter.

Live migration is not supported. An example of SR-IOV with VF passthrough is shown in Figure 2. Figure 2 is a simplified version of the hardware and software components.

Figure 2: Figure 2


Host
VM
VM
eth0
eth0
eth1
ENIC
ENIC
eth0
ENIC
Software layer
Hardware layer
pf0
vf0
vf1
vf2
Physical Functions
Virtual Functions
(vNICs)
305549