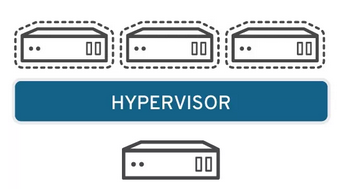
# 虚拟化技术

## 概述

虚拟化技术是将底层物理设备（CPU、内存、外设等）与上层操作系统、应用软件分离的一种解耦技术，在一台物理机上逻辑划分出多个虚拟机使用这些计算资源。

虚拟化的核心思想就是提高底层硬件资源的利用率，通过在操作系统中增加一个虚拟化层，将底层的资源池化，向上层操作系统提供一个预期一致的服务器硬件环境，并允许不同操作系统的虚拟机相互隔离并且运行在同一个物理机上。



KVM虚拟化主要包括CPU虚拟化、内存虚拟化、网络虚拟化、存储虚拟化。

其中，KVM虚拟机CPU的软件调休主要是借助NUMA技术；内存方面的调优手段主要是KSM，即相同的内存页合并、内存气球技术及大页内存的使用。这些都在后面专门详述。

## 历史

在虚拟化技术发展初始，主要应用于IBM的大型机中，随着X86服务器架构流行，VMware开发出第一个针对X86服务器的商用虚拟化技术。后来，剑桥大学一名讲师发布了开源虚拟化项目XEN。

## 分类

虚拟化类型一般分为两类：I型虚拟化和II型虚拟化。

### I型虚拟化

Hypervisor直接安装在物理机上，多个虚拟机在Hypervisor上运行。Hypervisor实现方式一般是一个特殊定制的Linux系统。Xen和VMWare的ESXi都属于这个类型。

### II虚拟化

物理机上首先安装常规的操作系统，比如RedHat、Ubuntu和Windows。Hypervisor作为OS上的一个程序模块运行，并对管理虚拟机进行管理。KVM、VirtualBox和VMWare Workstation都属于这个类型。

### ----

目前虚拟化技术，大致包括：软件模拟、虚拟化层翻译、容器虚拟化。

### 软件模拟

软件模拟是通过软件完全模拟CPU、内存、网卡、磁盘等计算机硬件设备，其优点在于可以模拟任何硬件设备，但是其效率非常低，不具备实际应用价值。

采用该技术的产品有QEMU。

### 虚拟化层翻译

Hypervisor是一种运行在物理服务器和操作系统之间的中间软件层，可允许多个操作系统和应用共享一套基础物理硬件，因此也可以看作是虚拟环境中的“元”操作系统，它可以协调访问服务器上的所有物理设备和虚拟机，也叫虚拟机监视器（Virtual Machine Monitor）。Hypervisor是所有虚拟化技术的核心，非中断地支持多工作负载迁移的能力是Hypervisor的基本功能。当服务器启动并执行Hypervisor时，它会给每一台虚拟机分配适量的内存、CPU、网络和磁盘，并加载所有虚拟机的客户操作系统。

该技术又分为三种类型：

1. 全虚拟化：通过虚拟化引擎VMM，捕获虚拟机的指令并处理。

说明：这里的处理不是完完整整的将虚拟机的指令传递给宿主机，因为虚拟机传递的是越级的指令，所以VMM需要做些额外处理工作，比如虚拟机执行reboot，宿主机不会重启。

采用该技术的产品有VMWare ESXi。

1. 半虚拟化：全虚拟化方式由于在虚拟化层做了大量的“翻译”工作，效率

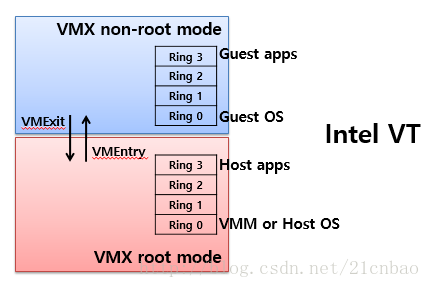
较低，所以XEN提出将虚拟化层的部分功能剥离，放到虚拟机内部处理，这就要求对虚拟机内核进行改造。

采用该技术的产品有XEN、Hyper-V。

1. 硬件支持的全虚拟化：前面的两种虚拟化方式都是在软件层面上进行的，

后来芯片厂商Intel推出了硬件解决方案，对CPU指令进行改造，即VT-x（AMD的硬件解决方案为AMD-v），其效率非常高。

VT-x增加了两种操作模式：VMX root模式和VMX non-root 模式，虚拟化引擎VMM运行在VMX root模式，虚拟机运行在VMX non-root模式，这两种操作模式都支持ring0~ring4这4个特权级。



对于虚拟机而言，Guest OS运行在non-root模式的ring0，其上的应用程序运行在non-root模式的ring3；而对于物理机而言，host OS和VMM则运行于root模式下的ring0，物理机的应用程序运行在root模式的ring3。Guest OS下发的越级指令，会VMExit到root模式下ring0内的VMM，从root环境到non-root环境叫做VMEntry，实际使用时在二者之间不断切换。

采用该技术的产品有KVM。

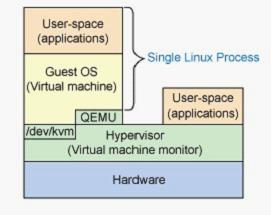
### 容器虚拟化

容器虚拟化基于CGroup和Namespace技术实现进程隔离的，每个进程有自己隔离的资源、根目录、进程号、内存空间等。但是，Docker在网络、磁盘上还受限制。

# KVM概述

KVM全称是Kernel-Based Virtual Machine，也就是说KVM是基于Linux内核实现的。

## KVM架构



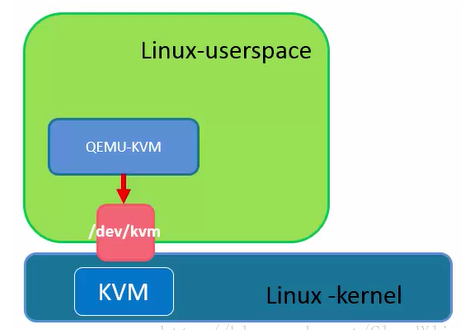
Guest：客户机系统，包括CPU（vCPU）、内存（vMem）、驱动（网卡、I/O 设备驱动等），被 KVM 置于一种受限制的CPU（即non-root）模式下运行。

KVM：运行在内核空间，提供CPU和内存的虚级化，以及客户机的I/O拦截。Guest的I/O被KVM拦截后，交给QEMU处理。

QEMU：修改过的为KVM虚机使用的QEMU代码，运行在用户空间，提供硬件I/O虚拟化，通过ioctl /dev/kvm设备和KVM交互。

## QEMU与KVM

QEMU是一个硬件模拟器，内建HDCP、DNS、TFTP等服务器，但是其速度慢实用价值不大。而KVM只是内核的一个模块，它没有用户空间的管理工具，KVM的虚拟机可以借助QEMU的管理工具管理，QEMU也可以借助KVM实现加速。现在，KVM与QEMU已经结合在一起。



/dev/kvm是KVM内核模块提供给用户空间的一个接口，这个接口被qemu-kvm调用，通过ioctl系统调用就可以给用户提供一个工具用于管理虚拟机生命周期。qemu-kvm就是通过open()、close()、ioctl()等方法打开、关闭、调用这个接口实现与KVM交互的。

## libvirt与KVM

libvirt是Redhat开源的虚拟化管理工具，除了能够管理KVM这种Hypervisor，还可以管理Xen、VirtualBox等。

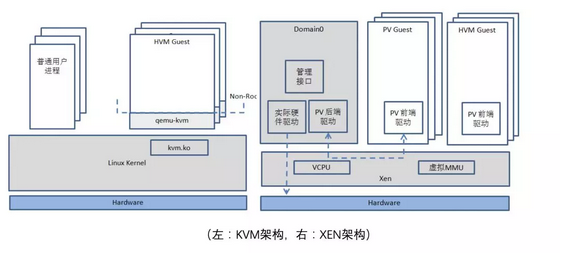
主要有3部分组成：

1. lib库：支持C、Python、Ruby等编程语言；
2. libvirtd服务
3. virsh命令行工具

libvirt可以管理不同的虚拟化引擎，但是目前实际应用环境中只有KVM在使用。可以实现虚拟机的生命周期管理以及外设的添加。

## KVM与XEN

KVM较XEN出现的晚，但是由于XEN需要对内核补丁，稍微不慎，系统崩溃，加之KVM是Linux的一部分，可以使用通常的Linux调度器和内存管理，这使得KVM更小更易使用。



# 环境搭建

## CPU检测

egrep -c '(vmx|svm)' /proc/cpuinfo

输出结果大于0表示支持虚拟化，如果等于0则需要手动选择支持虚拟化。

## 安装KVM

yum install qemu-kvm

安装虚拟化管理工具：

yum install virt-manager libvirt libvirt-python python-virtinst bridge-utils

## 加载kvm.ko

modprobe kvm

查看驱动是否加载成功：

lsmod | grep kvm

如果加载成功会有输出信息，没有加载成功grep不到指定驱动信息。

## 启动libvirt

service libvirtd restart

chkconfig libvirtd on

## 创建系统盘

qemu-img create -f qcow2 /kvmtest/centos-7.qcow2 10G

## 启动虚拟机

virt-install --virt-type kvm --name centos-7 --ram 1024 \

--vcpus 1 \

--cdrom=/kvmtest/CentOS-7-x86\_64-minimal.iso \

--disk /kvmtest/centos-6.8.qcow2,format=qcow2 \

--network network=default \

--graphics vnc,listen=0.0.0.0 --noautoconsole \

--os-type=linux --os-variant=rhel6

各参数的含义如下：

-n NAME 指定虚拟机的名称

-r MEMORY 指定虚拟机的内存用量

-u UUID 指定虚拟机的唯一通用标识符（UUID）。

省略这个参数时，virt-install 将会自动产生

–vcpus=VCPUS 指定虚拟机的虚拟 CPU （Virtual CPU，VCPU） 数量

-f DISKFILE 指定虚拟磁盘的文件路径名称

-s DISKSIZE 用来指定虚拟磁盘的大小，这个参数需配合-f 使用。

DISKSIZE为虚拟磁盘的大小，单位是GB

-m MAC 指定虚拟机的网络卡之硬件地址。

这个参数可以省略，省略时virt-install 将自动产生

-p 以半虚拟化的方式建立虚拟机

-l LOCATION 指定安装来源

## VNC连接虚拟机

netstat -natp

# CPU虚拟化

从计算层面来说，X86物理服务器上的CPU、内存资源都可提供给虚拟机使用.现在的高性能X86服务器一般都是多CPU多核系统，NUMA 架构会越来越受欢迎，因为这个架构可解决多处理器多核和非统一内存架构之间的交互带来的新的CPU、内存资源分配方法的挑战，并改善占用大量内存的工作负载的性能。

通俗来讲，CPU虚拟化就是用单个CPU模拟出多个并行的CPU，允许一个平台同时运行多个操作系统，并且应用程序都可以在相互独立的空间内运行而互不影响，从而显著提高计算机的工作效率。

CPU的虚拟化技术是一种硬件方案，原因是纯软件虚拟化解决方案存在很多限制 。“客户”操作系统很多情况下是通过VMM(Virtual Machine Monitor，虚拟机监视器)来与硬件进行通信，由VMM来决定其对系统上所有虚拟机的访问。(注意，大多数处理器和内存访问独立于VMM，只在发生特定事件时才会涉及VMM，如页面错误)。在纯软件虚拟化解决方案中，VMM在软件套件中的位置是传统意义上操作系统所处的位置。如处理器、内存、存储、显卡和网卡等)的接口，模拟硬件环境。这种转换必然会增加系统的复杂性。支持虚拟技术的CPU带有特别优化过的指令集来控制虚拟过程，通过这些指令集，VMM会很容易提高性能，相比软件的虚拟实现方式会很大程度上提高性能。

虚拟化技术可提供基于芯片的功能，借助兼容VMM软件能够改进纯软件解决方案。由于虚拟化硬件可提供全新的架构，支持操作系统直接在上面运行，从而无需进行二进制转换，减少了相关的性能开销，极大简化了VMM设计，进而使VMM能够按通用标准进行编写，性能更加强大。另外，在纯软件VMM中，目前缺少对64位客户操作系统的支持，而随着64位处理器的不断普及，这一严重缺点也日益突出。而CPU的虚拟化技术除支持广泛的传统操作系统之外，还支持64位客户操作系统。

虚拟化技术是一套解决方案。完整的情况需要CPU、主板芯片组、BIOS和软件的支持，例如VMM软件或者某些操作系统本身。即使只是CPU支持虚拟化技术，在配合VMM的软件情况下，也会比完全不支持虚拟化技术的系统有更好的性能。

Intel处理器产品线中应用英特尔Virtualization Technology(Intel VT)虚拟化技术，而AMD支持AMD Virtualization Technology(AMD VT)虚拟化技术。

# 内存虚拟化

## 概述

通过内存虚拟化共享物理系统内存，动态分配给虚拟机。现在所有的X86 CPU都包括了一个称为内存管理的模块MMU（Memory Manager Unit）和TLB（Translation Lookaside Buffer），通过MMU和TLB来优化虚拟内存的性能。

KVM 中，虚机的物理内存即为 qemu-kvm 进程所占用的内存空间。KVM 使用 CPU 辅助的内存虚拟化方式。在 Intel 和 AMD 平台，其内存虚拟化的实现方式分别为：

AMD 平台上的 NPT （Nested Page Tables） 技术

Intel 平台上的 EPT （Extended Page Tables）技术

EPT 和 NPT采用类似的原理，都是作为 CPU 中新的一层，用来将客户机的物理地址翻译为主机的物理地址。

## KSM

## 内存气球

## 内存限制

## 巨页

# 网络虚拟化

# 磁盘虚拟化

## 概述

QEMU作为磁盘设备的软件模拟，后来结合KVM技术，加上Virtio半虚拟化技术的支持，性能得到很大改善。

KVM虚拟机可以指定IDE、SATA、Virtio、Virtio-SCSI几种磁盘格式，CentOS6.x只支持IDE和VIrtio磁盘类型，CentOS7.x增加支持SATA和Virtio-SCSI磁盘类型。IDE和SATA是全虚拟化的磁盘，性能较低；Virtio和Virtio-SCSI是半虚拟化磁盘，通过改造虚拟机的驱动达到目标。

## 应用场景

## 磁盘缓存

### 概述

在配置磁盘的时候，可以指定缓存模式。每个虚拟机的磁盘接口可以配置成writethrough、writeback、none、direstsync或者unsafe缓存模式。

### 性能

### 应用

## 磁盘镜像格式

### 磁盘镜像文件

### 裸设备

### 性能对比

### 应用场景

# virsh指令

## 创建虚机

如果希望借助virt-manager通过复制的方式创建虚机，可以采用如下操作：

1. 通过virt-manager命令打开KVM虚拟机管理图形化界面
2. 选择要复制的虚拟机，右键shut down-force off，将虚拟机关闭
3. 选择要复制的虚拟机，右键clone，弹出窗口，修改虚机名称
4. 修改镜像名称
5. 创建好虚拟机
6. 在virt-manager图形化界面上选中新建的虚拟机，点击open
7. 依次选中Detail-Overview，然后可以修改虚机名称，点击Apply
8. 依次点击Add Hardware，选中Storage，选中SELECT managed or other existing storage，点击浏览选中已经复制的镜像文件，点击close volume，点击完成
9. 修改网卡信息

## 删除虚机

通过virt-manager方式删除KVM虚拟机：

1. 通过virt-manager命令打开KVM虚机图形化管理界面
2. 选择要删除的KVM虚拟机，右键shutdown-force off关闭虚机
3. 选择要删除的虚机，右键delete

如果想彻底删除10.47.181.1这个虚拟机，需要执行以下操作：

1. virsh destroy 10,47,181,1停止虚拟机
2. virsh list –all确认该虚拟机已经处于关闭态
3. virsh undefine 10.47.181.1解除标记
4. 通过查询/etc/libvirt/qemu/10.47.181.1.xml配置文件，找到镜像文件位置/var/lib/libvirt/images/10.47.181.1.qcow2
5. 删除虚机文件：

rm –rf /var/lib/libvirt/images/10.47.181.1.qcow2

rm –rf /etc/libvirt/qemu/10.47.181.1.xml

1. virsh list –all观察10.47.181.1已经被删除

## 修改虚机名

如果想要修改虚拟机名称，可以采取如下操作：

1. 关闭虚拟机：virsh shutdown vm-1
2. 导出xml文件：virsh dumpxml vm-1 > vm-2.xml
3. 编辑vm-2.xml，修改vm名称
4. 重定义：

virsh undefined vm-1

virsh define vm-2.xml

1. 启动虚拟机：virsh start vm-2
2. 查看：virsh list

## 打开虚机问题

在使用virt-manager打开KVM虚拟机输入用户名时键盘无法使用的问题，解决办法如下：

1. 使用virt-manager打开虚机管理界面，右键open，点击run虚机，进入登录界面，点击Not Found进入输入用户界面
2. 此时出现无法输入问题，先将此窗口关闭，然后再关闭的虚拟机上右键Shut down Force Off，将虚机关闭
3. 在图形化界面上选中关闭的虚拟机，点击打开Detail Display Spice，Type选择VNC Server，Keymap选择en-us，点击应用即可