<http://blog.csdn.net/qiqijianglu/article/details/14451809>

全虚拟化(Full virtualization), 也称为原始虚拟化技术, 是另一种虚拟化方法. 该模型使用虚拟机协调客户[操作系统](http://product.it168.com/list/b/0501_1.shtml)和原始硬件(见图2). 这里"协调"是一个关键词, 因为VMM在客户操作系统和裸硬件之间用于工作协调. 一些受保护的指令必须由Hypervisor(虚拟机管理程序)来捕获和处理. 因为操作系统是通过Hypervisor来分享底层硬件.

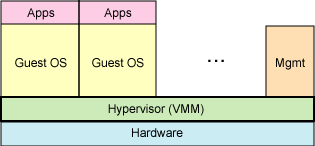


　　图2. 全虚拟化: 使用Hypervisor分享底层硬件

　　全虚拟化的运行速度要快于硬件模拟, 但是性能方面不如裸机, 因为Hypervisor需要占用一些资源. 全虚拟化最大的优点是操作系统没有经过任何修改. 它的唯一限制是操作系统必须能够支持底层硬件(比如, PowerPC).

**老机器上的Hypervisors**

　　一些老的硬件如x86, 全虚拟化遇到了问题. 比如, 一些敏感的指令需要由VMM来处理(VMM不能设置陷阱). 因此, Hypervisors必须动态扫描和捕获特权代码来处理问题.

**半虚拟化**

　　半虚拟化(Paravirtualization)是另一种类似于全虚拟化的热门技术. 它使用Hypervisor(虚拟机管理程序)分享存取底层的硬件, 但是它的客户操作系统集成了虚拟化方面的代码. 该方法无需重新编译或引起陷阱, 因为操作系统自身能够与虚拟进程进行很好的协作.

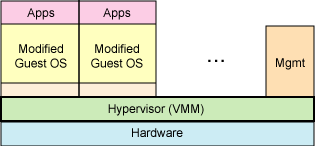


　　图3. 半虚拟化: 通过客户操作系统分享进程

　　上面提到过, 半虚拟化需要客户操作系统做一些修改(配合Hypervisor), 这是一个不足之处. 但是半虚拟化提供了与原始系统相近的性能. 与全虚拟化一样, 半虚拟化可以同时能支持多个不同的操作系统.

有这样的疑问：

在半虚拟化的环境中，不能运行未经修改内核的操作系统，那么，cpu不支持vt，在宿主系统上跑的虚拟环境就可以称为半虚拟化环境了，那么在cpu不支持vt的机器上vmwera可以运行xp，这个互相矛盾，为什么就可以运行呢？

PV vs. FV  
PV（Para-Vritralization）和FV（Full-Vritralization）的差别，主要以guest OS的硬件仿真程度做区分。  
  
FV：FV是一般较常看到的作法，所有的guest OS完全不会看到实际的硬件为何，只能使用由Supervisor所提供的所有虚拟硬件，因此，在这种机制下，guest OS动作的性能一定会大受虚拟接口的影响。另外还有一个特点，就是因为完全仿真的关系，不支持新的技术，连ACPI开关机的机制都无法使用，也就是当使用者在FV的guest OS下，若直接触动关机的按钮（这里的按钮是由VMM所提供的，不是主机上的）会直接断电，而不会进行关机程序。  
  
PV：至于PV的作法，有鉴于一般Virtual Machine工具都是以完全仿真的方式，造成性能上的降低，因此，XEN在设计上，希望各操作系统可以在开发时就已经将XEN的技术包括进去，这样在使用时，就可以用局部仿真的方式，让操作系统可以直接使用到硬件中的CPU、内存等，而不需要通过XEN做仿真的操作。  
  
   
  
这样，若硬件都是由Virtual Machine仿真出来的，性能自然会变得比较慢，所以XEN所主推的概念就是，当操作系统默认支持XEN时就可以通过XEN的机制，直接使用到底层的硬件，而不是每个OS都要通过Hypervisor的接口，性能上就可大为提高。  
  
在PV与FV的安装上，对CPU的要求是有差异的。以PV而言，因为OS原本就支持XEN，所以不需要经过特殊的处理步骤，因此，对CPU来于，没有特殊的要求，基本上都可以使用。

但FV就不同，因为FV模式是以完全仿真的方式进行，所以CPU必须要特别将此功能加入到CPU的核心中，因此，CPU是有特殊需求的。在Intel与AMD中针对Virtual Machine所加入的功能名称不同，分别为Intel VT与AMD-V。  
  
在目前Linux的世界，XEN并不是唯一的，除了XEN这一架构外，另外还有一个KVM（Kernel based Vitrual Machine）,从2.6.20版本之后的Linux Kernel就支持其功能。因为KVM的做法是以原本Kernel为依据，只要将其module加载即可使用，唯一较XEN弱势的是KVM以单一Full Virtuallization为服务方式。

虚拟化最近几年取得了长足的进步，首先由于有大量的开源虚拟机管理程序在不断发展，这个改进几乎排除了操作系统与大功率服务器快速增长的利用之间的障碍，公司立即受益。知道最近，软件仿真虚拟化一直是关注的焦点，两个常见的软件仿真虚拟化途径是完全虚拟化和半虚拟化。对于完全虚拟化，通常调用管理程序或虚拟机监视器，存在于虚拟操作系统和硬件层之间，这个层在操作系统实例之间多路复用竞争系统资源。半虚拟化不同，管理程序在一个协作的环境中运作，因为每个客户操作系统都知道它正运行在一个虚拟化环境中，因此每个使用管理程序的协作都在硬件下完成虚拟。

两个途径都有优势也有劣势，半虚拟化的主要优势在于它运行尽可能快地基于软件的虚拟化，以不支持有专利的操作系统为代价。完全虚拟化没有这个限制，但是，完全虚拟化管理程序非常复杂。VMware，一个商业虚拟化解决方案，它就是一个完全虚拟化的例子，Xen，用户模式Linux(UML)和其他提供的是半虚拟化解决方案。

简单介绍一下基于硬件的虚拟化，这个线路已经变得模糊起来，随着Intel的VT技术和AMD的SVM技术的出现，编写一个管理程序变得更加简单了，现在，保持管理程序复杂性到最小程度，对于完全虚拟化受益匪浅。

Xen作为最优秀的半虚拟化引擎，在基于硬件的虚拟化的帮助下，现在也支持完全虚拟化MS windows了。KVM是一个相对较新的简单的，但也非常强大的虚拟化引擎，它已经集成到Linux内核中去了，让内核天生有虚拟化的能力，因为KVM使用的是基于硬件的虚拟化技术，它不需要修改客户操作系统，因此，部署在一个受支持的处理器上，它可以从Linux支持任何平台。

一、Xen是一款虚拟化软件，支持半虚拟化和完全虚拟化。它在不支持VT技术的cpu上也能使用，但是只能以半虚拟化模式运行。  
二、半虚拟化的意思是需要修改被虚拟系统的内核，以实现系统能被完美的虚拟在Xen上面。完全虚拟化则是不需要修改系统内核则可以直接运行在Xen上面。  
三、VMware是一款完全虚拟化软件。完全虚拟的弱点是效率不如半虚拟化的高。半虚拟化系统性能可以接近在裸机上的性能。  
四、 Xen是由一个后台守护进程维护的，叫做xend，要运行虚拟系统，必须先将它开启。它的配置文件在/etc/xen/xend-config.sxp，内容包括宿主系统的类型，网络的连接结构、宿主操作系统的资源使用设定，以及vnc连接的一些内容。（如果你想增加一个虚拟网络设备的话，是需要在这里设定的）  
五、/etc/xen/auto 的含义是如果你想让被虚拟系统随着宿主系统一同启动的话，就把虚拟系统的配置文件放到这个目录下面来。  
六、/etc/xen/scripts 是些脚本文件，用于初始化各种虚拟设备，比如虚拟网桥等。（如果要增加一个虚拟网络设备，同样需要在此处调节）  
七、在/etc/xen下面会有些配置文件，这就是虚拟系统引导时所必须的些文件，里面记录了引导和硬件信息。  
八、Xen的配置工具有许多，我使用的是virt-manager（GUI）、virt-install和xm。第一个用于管理和安装系统，第二个只用于安装系统，第三个用于启动系统。  
九、安装半虚拟Linux有两种方法，一种是利用Linux的网络安装方式安装，http、ftp、nfs方式都是可以的（特别注意：半虚拟环境下安装Linux是不支持本地光驱或者iso镜像安装的！），并且RHEL5会自动生成配置文件。第二种是先建立镜像文件，并格式化，然后挂载到本地文件系统上来，将虚拟系统需要用到的文件拷贝进去并修改，然后手工创建配置文件并启动。  
十、虚拟网络设备有三种模式：bridge桥模式、router路由模式和nat模式。其中桥模式是默认模式，在这种模式下，虚拟系统和宿主系统被认为是并列的关系，虚拟系统被配置IP或者dhcp后即可联通网络。  
十一、原来的iptables无法对桥模式下的数据包做处理，RHEL5的iptables中增加了一个physdev的模块，可用iptables -m physdev -h查看帮助