## CPU虚拟化：

--基于二进制翻译技术（Binary Translation）的完全虚拟化。 假如操作系统试图去执行指令A，但该指令A会给虚拟化引 擎带来某些问题，那么BT将把它转换成某些更合适的指令并 伪造指令A应该返回的结果。这是一项欺骗工作，且占用大 量CPU资源，另外用许多代码取代一条代码也不会使事情运 作得快些。

--paravirtualization半虚拟化

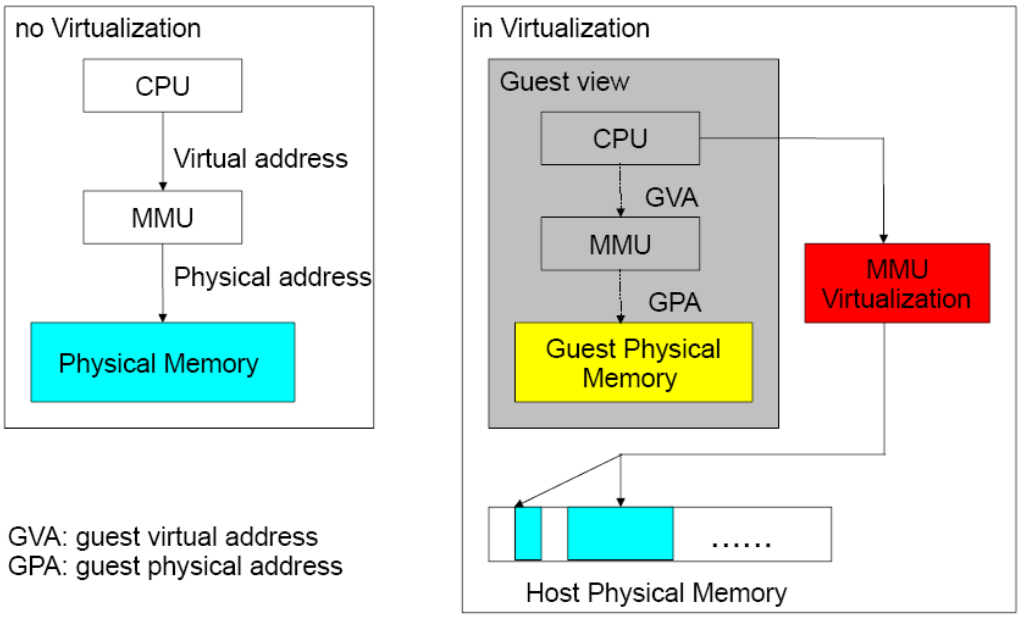
--硬件辅助虚拟化 Hardware-assistant Virtualization Mechine HVM

\*Intel--VT-x \*AMD--v 这两种技术通过将hostOS运行 于-1环，guestOS运行于0环，来解决运行敏感指令问题。

## **内存虚拟化**：

Intel和AMD分别通过EPT(Extended Page Tables)和 NPT(Nested Page Tables)为虚拟化应用提升影子MMU的性能， 并通过标记(tagged)TLB来避免虚拟机切换时频繁清写 (flush)TLB以提高TLB缓存的命中率。

注：TLB用来缓存线性地址--物理地址的映射，每个虚拟机 的同 一线性地址可能会映射到不同的物理地址，因此，虚拟机 切换时需要flushTLB。而tagged则通过对TLB多增加一个键 值，也就是一个标记位来标识映射关系所对应的虚拟机。

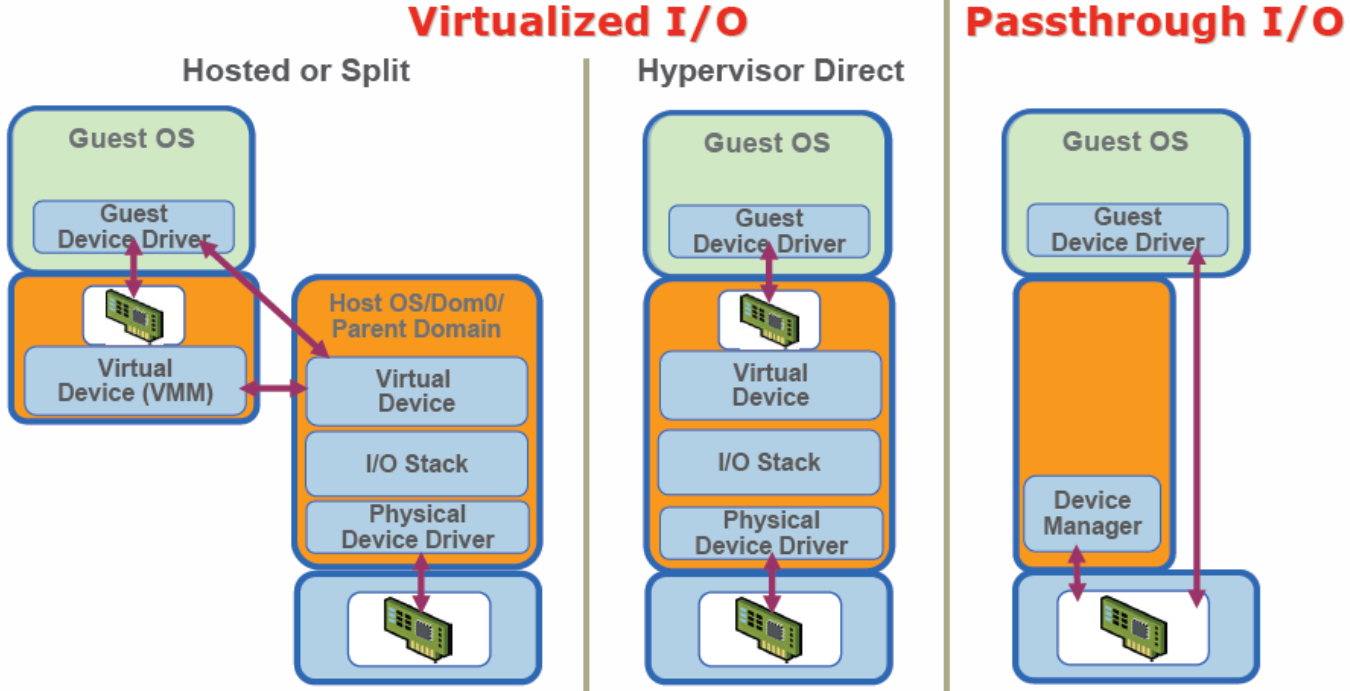


## IO虚拟化：

--模拟emulated

--半虚拟化

--passthough IO透传



虚拟化模型：

1 完全虚拟化

长方形：设备（虚拟/实际）

圆形：监控管理进程

三角形：设备驱动

kernel

Guest

Guest

VMM(hypervisor)

硬件

vmm是一个特制的精简的OS，利用CPU虚拟化中的BT或者HVM技术。IO虚拟化中的emulated技术。hypervisor可以捕获CPU指令，为指令访问硬件控制器和外设充当中介。因而，完全虚拟化技术几乎能让任何一款操作系统不用改动就能安装到虚拟服务器上，而它们不知道自己运行在虚拟化环境下。主要缺点是，hypervisor给处理器带来开销。

VMvare公司注重于完全虚拟化。

2 半虚拟化

guestOS

hypercall

VMM（hypervisor）

硬件

每个guestOS都“知道”自己所处的环境就是一个虚拟机，因此guestos的内核运行在ring3级别中。当硬件支持HVM时，也可以执行pv on HVM。即cpu 内存使用HVM，IO使用半虚拟化技术。

Xen是半虚拟化技术的领航者。

3 OS-level的虚拟化

VMM

kernel

硬件

所有guestos共用一个内核，guestos实际上就是运行于用户空间的一个进程。该技术比之半虚拟化技术性能更好，但若kernel出现故障则所有虚拟机无法运行。

该技术代表：container