**IO虚拟化**

**一些有用的知识**

**MMU(Memory Management Unit, 内存管理单元)**: 将虚拟地址转化为物理地址，cpu与主存之间地址转换。

**DMA(Direct Memory Access，直接内存存取)**: 是一种快速传送数据的机制。DMA 传输将数据从一个地址空间复制到另外一个地址空间。包括以下几

当CPU 初始化这个传输动作，传输动作本身是由 DMA 控制器来实行和完成。最初的DMA地址是物理地址，后来dmar就出现了。 dmar意为DMA remapping，是Intel为支持虚拟机而设计的I/O虚拟化技术，I/O设备访问的DMA地址不再是物理内存地址，而要通过DMA remapping硬件进行转译，DMA remapping硬件会把DMA地址翻译成物理内存地址，并检查访问权限等等。负责DMA remapping操作的硬件称为IOMMU。

**IOMMU**把设备访问的虚拟地址转化为物理地址，连接DMA-capable 总线和主存。为了防止设备错误地访问内存，有些IOMMU还提供了访问内存保护机制(检查访问权限)。IOMMU不仅将DMA地址虚拟化，还起到隔离，保护等作用。

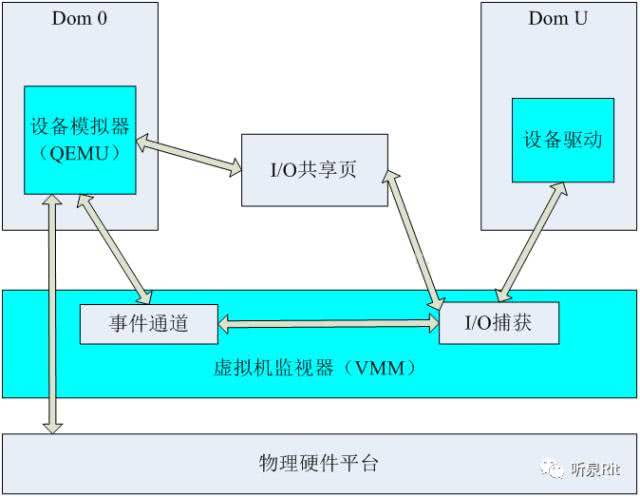
IOMMU还支持**中断重映射**：当设备发送了中断号的消息时，消息首先抵达IO MMU，通过中断映射表转换为一个新的中断，目标是正在运行指定虚拟机的CPU。

三种常见的IO虚拟化及其优缺点

常见的IO虚拟化主要有三种方案：**基于软件模拟的方案**，这种方案里, 中断、DMA的访问都是通过软件实现的，优点是可以模拟任何硬件的模型，缺点是性能不会太好；**半虚拟化技术**，主要是为了解决软件模拟性能问题，比如串口对性能要求不高可以采用软件模拟，但是磁盘设备、网卡设备对性能要求高，主流方案是采用半虚拟化技术，前后端相互感知，通过Shared Memory控制请求的传输，两个设备之间的通知也是基于快速消息传递，性能很高；**设备直通模式**，比如PCIE的直通、网卡SROV直通，对性能更高的可以采用此模式，可以达到和物理机上直接使用接近的性能，但是设备和虚拟机的耦合会对管理造成影响

**IO全虚拟化**

即通过 VMM 模拟 I/O 设备（磁盘和网卡等）实现虚拟化。这种模式下，Guest OS 所能看到的就是一组统一的 I/O 设备。VMM 截获 Guest OS 对 I/O 设备的访问请求，然后通过软件模拟真实的硬件。这种方式对 Guest OS 而言非常透明，无需考虑底层硬件的情况。



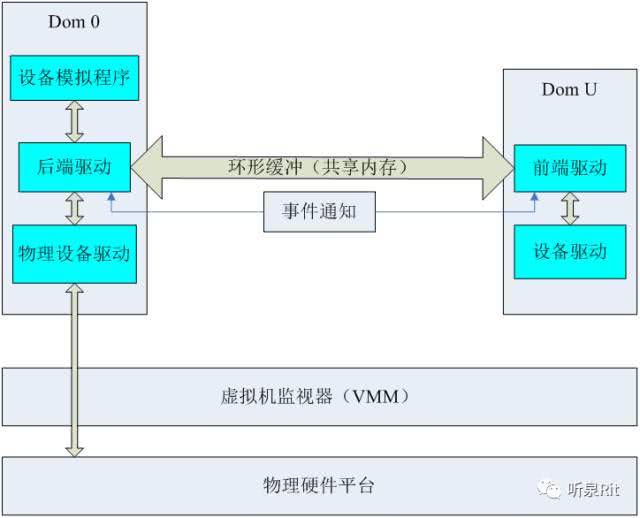
其最大的优点在于不需要对操作系统内核做修改，也不需要为其改写驱动程序，因此，这是可移植性与兼容性最佳的一种 I/O 设备虚拟模型，这也是它被如此广泛使用的主要原因。但是 I/O 全虚拟化有一个很大的不足之处，就是性能较差，主要原因有两方面：

（1）第一、模拟方式是用软件行为进行模拟，这种方式本身就无法得到很高的性能；

（2）第二、这种模型下 I/O 请求的完成需要虚拟机与监视器程序多次的交互，产生大量的上下文切换，造成巨大开销。

****I/O 半虚拟化技术****

即前端（Front-End）和后端（Back-End）模型，模拟实现虚拟化。如下图所示：



****其中，Guest OS 中的驱动程序为前端，VMM 提供的与 Guest 通信的驱动程序为后端****。

前端驱动将 Guest OS 的请求通过与 VMM 间的特殊通信机制和接口发送给 VMM 的后端驱动，后端驱动对 VM 的数据进行分时分通道处理，处理完请求后再发送给物理驱动。

该模型采用了 I/O 环机制，减少了虚拟机与虚拟机监视器之间的切换；同时该模型摒弃了传统的中断机制，而采用事件或回调机制来实现设备与客户机间的通信。进行中断处理时，传统的中断服务程序需要进行中断确认和上下文切换，而采用事件或回调机制，无需进行上下文切换。

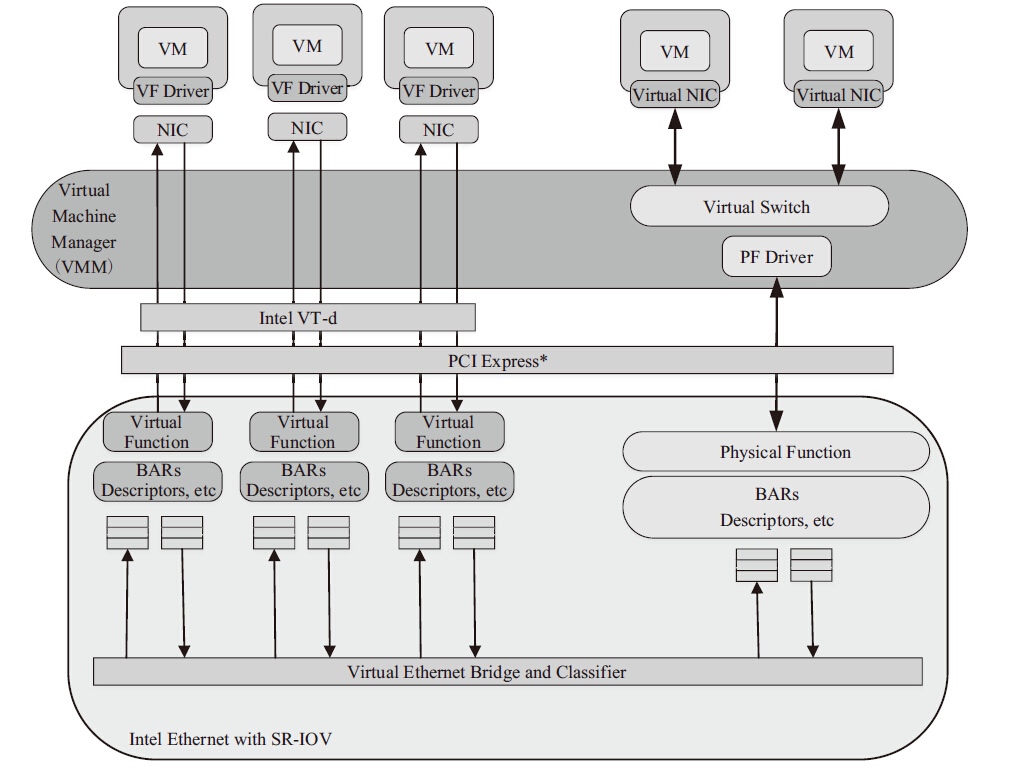
I/O 半虚拟化模型虽然在性能上比 I/O 全虚拟化模型要好，但是这种 I/O 模型有一个很大的缺点，就是要修改操作系统内核以及驱动程序，因此会存在移植性和适用性方面的问题，导致其使用受限。

**SR-IOV 单根I/O虚拟化**

SR-IOV技术是由PCI-SIG制定的一套硬件虚拟化规范，全称是Single Root IO Virtualization（单根IO虚拟化）。SR-IOV规范主要用于网卡（NIC）、磁盘阵列控制器（RAID controller）和光纤通道主机总线适配器（Fibre Channel Host Bus Adapter，FC HBA），使数据中心达到更高的效率。SR-IOV架构中，一个I/O设备支持最多256个虚拟功能，同时将每个功能的硬件成本降至最低。SR-IOV引入了两个功能类型：

**PF**（Physical Function，物理功能）：这是支持SR-IOV扩展功能的PCIe功能，主要用于配置和管理SR-IOV，拥有所有的PCIe设备资源。PF在系统中不能被动态地创建和销毁（PCI Hotplug除外）。

**VF**（Virtual Function，虚拟功能）：“精简”的PCIe功能，包括数据迁移必需的资源，以及经过谨慎精简的配置资源集，可以通过PF创建和销毁



一旦在 PF 中启用了 SR-IOV，就可以通过 PF 的总线、设备和功能编号（路由 ID）访问各个 VF 的 PCI 配置空间。每个 VF 都具有一个 PCI 内存空间，用于映射其寄存器集。VF 设备驱动程序对寄存器集进行操作以启用其功能，并且显示为实际存在的 PCI 设备。创建 VF 后，可以直接将其指定给 I/O 来宾域或各个应用程序。此功能使得虚拟功能可以共享物理设备，并在没有 CPU 和虚拟机管理程序软件开销的情况下执行 I/O。