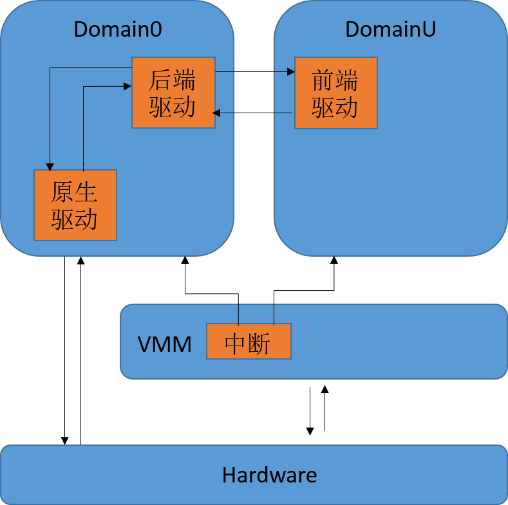
I/O虚拟化

一、概要

利用Xen架构，以下简述一个简单的I/O虚拟化架构。

架构分为三个部分：Hardware、VMM、Domain

Domain：虚拟机部分，有一个Domain0 和 DomainU组成。其中Domain0为管理者，管理其他DomianU。

VMM：虚拟机与硬件之间的软件层，负责CPU调度等非I/O操作的事务，提供接口，其中包括中断机制。

二、I/O过程

1.虚拟机->硬件

(1)前端驱动：DomainU中的硬件驱动为前端驱动，实际不存在任何硬件驱动功能，可以认为是一个简单的文件，记录所有的I/O操作。

(2)后端驱动：存在于Domain0中，主要作用为收集DomainU的前端驱动中的所有I/O操作，解析I/O地址，同时根据各个DomainU中的实际操作系统的I/O操作接口，转化为Domain0系统的I/O接口，之后交给原生驱动。

(3)原生驱动：真实的驱动，可以与硬件直接交互，接受后端驱动的I/O操作后会发送给Hardware执行。

2．硬件->虚拟机

硬件完成相应的操作后会发送中断给VMM，由VMM将中断转化为虚拟中断发送给Domain0和相应的DomainU，通知其I/O操作完成。然后DomainU从Domain0中获取相应的数据。

三、效率

为了加快Domain0和DomainU之间的数据传递，采用共享内存的方式，需要VMM的内存管理支持。当DomainU发出I/O请求时，会将一个空的内存页同时发给后端驱动，后端驱动会将I/O结果直接写到此内存页中（此页不单单用作写结果，例如DMA，此时会利用这个页来完成I/O操作），之后DomainU访问此页获取结果。

为了完善以上功能，同时增强各个虚拟机之间的信息传递，为每个Domain维护一个表，表中记录各个页面可以被哪些Domain访问（典型在发出I/O请求时需要设置一个页面可以被Domain0使用）。通过此表可以方便I/O地址的转换，因为此时I/O地址即为设置的内存页。