



翱翔在Linux的天空

HumJb & HaHa

[首页](#) | [博文目录](#) | [关于我](#)

humjb_1983

博客访问: 9496

博文数量: 80

博客积分: 0

博客等级: 民兵

技术积分: 685

用户组: 普通用户

注册时间: 2014-02-20 08:27

[加关注](#)[短消息](#)[论坛](#)[加好友](#)

文章分类

[全部博文 \(80\)](#)[硬件相关 \(5\)](#)[虚拟化 \(13\)](#)[其他 \(1\)](#)[Linux其他方面 \(3\)](#)[Linux内核 \(57\)](#)[未分配的博文 \(1\)](#)

文章存档

[2014年 \(80\)](#)

我的朋友



asuka20



321leon

最近访客



arm-linu



码出一片



pisming



jeppeter



刘一痕



SCvsCS

KVM基本原理及架构三-CPU虚拟化

2014-04-10 12:38:08

分类: LINUX

3 CPU虚拟化

3.1 基本原理

CPU虚拟化是VMM中最核心的部分, 由于内存和IO访问的指令都是敏感指令, 所以内存和IO虚拟化都依赖于CPU虚拟化的实现。CPU虚拟化的目标是让虚拟机中执行的所有敏感指令都能产生异常而“陷入”, 并由VMM进行模拟。VMM的陷入是通过CPU的保护机制、中断或异常来完成的。通常, VMM的陷入方式有如下3种:

- 1、由CPU的保护机制触发。CPU在执行敏感指令之前, 会检查执行条件是否满足, 执行条件主要包括: 当前特权级别、运行模式、内存映射关系等, 只要有任一条件不满足, 就会VM-Exit陷入到VMM进行进一步处理。
- 2、异步中断。包括处理器内部的中断源和外设的中断源, 当中断信号到达CPU时, CPU会强行中断Guest OS当前执行的指令, 然后VM-Exit到VMM注册的中断服务程序进行进一步处理。
- 3、虚拟机主动触发的异常, 也就是通常所说的陷阱。虚拟机可以通过陷阱指令主动VM-Exit到VMM中。

在KVM虚拟化解决方案中, 利用了硬件的虚拟化特性(比如Intel VT-x和AMD SVM)实现CPU的虚拟化。如之前介绍, VT-x提供了一套称作VMX的新的工作模式, 工作在该模式下的处理器又具有两类操作模式: VMX root operation和VMX non-root operation。对操作系统来说, VMX non-root operation模式与传统的x86处理器兼容, 最大的差别在于当虚拟机执行一些访问全局资源的指令时将导致虚拟机退出操作 (VM-Exit), 从而使虚拟机监控器获得控制权, 以便对访问全局资源的指令进行模拟。以后, 虚拟机监控器可以通过虚拟机进入操作 (VM-Entry) 使虚拟机重新获得控制权。

其次, VT-x为系统编程接口状态的切换提供硬件支持。VT-x为每个虚拟机维护至少一个VMCS (Virtual Machine Control Structure) 结构, 其中保存了虚拟机和虚拟机监控器的系统编程接口状态。当执行VM exit和VM entry操作时, VT-x自动根据VMCS中的内容完成虚拟机和虚拟机监控器间的系统编程接口状态切换。另外, VT-x还提供了一组指令, 使得虚拟机监控器通过一条指令就可以完成虚拟机间的切换。

3.2 VCPU


硬件虚拟化使用VCPU(Virtual CPU)描述符来描述虚拟CPU, VCPU描述符与OS中进程描述符类似, 本质是一个结构体, 其中包含如下信息:

- ü VCPU标识信息, 如VCPU的ID号, VCPU属于哪个Guest等。
- ü 虚拟寄存器信息, 在VT-x的环境中, 这些信息包含在VMCS中。
- ü VCPU状态信息, 标识白VCPU当前所处的状态(睡眠、运行等), 主要供调度器使用。
- ü 额外的寄存器/部件信息, 主要指未包含在VMCS中的寄存器或CPU部件, 比如: 浮点寄存器和虚拟的LAPIC等。
- ü 其他信息: 用户VMM进行优化或存储额外信息的字段, 如: 存放该VCPU私有数据的指针。


总体来说, VCPU可以划分为两部分:

- 1、以VMCS为主, 由硬件来使用和更新的部分, 主要是虚拟寄存器。
- 2、除VMCS之外, 由VMM来使用和更新的部分。如VCPU标识、状态信息等。


当VMM创建虚拟机时, 首先要为虚拟机创建VCPU, 整个虚拟机的运行实际上可以看做VMM调度不同的VCPU运行。



风铃之音



embedde



chrxy

订阅

推荐博文

·云计算-Azure-3.负载均衡集...

·读书与写论文的引导书——leo...

·在framework层添加自己的jar...

·tcpdump工具浅析

·python json ajax django四星...

·Solaris文件管理和目录管理...

·Solaris退出系统,改变系统运...

·监控Data Guard实时同步...

·Oracle的告警日志之v\$diag_al...

·使用AWR生成报表

热词专题

·Debian设置

·欢迎kkkkkkkybbb在ChinaUnix...

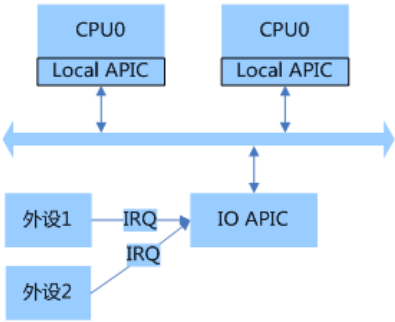
·虚拟机ping不通win7宿主机...

·安装oracle

·关于STM32的SPI的问题

3.3 中断虚拟化

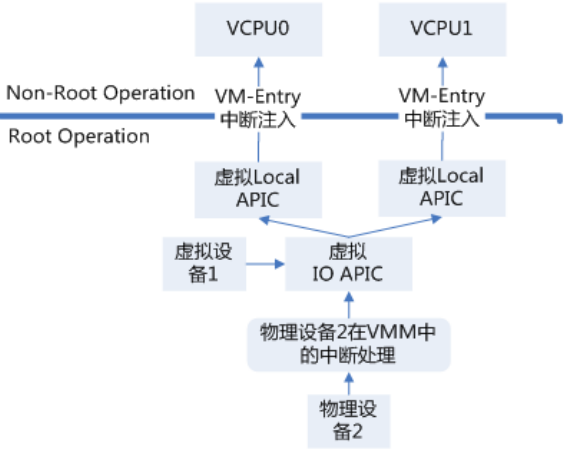
在物理平台上，中断架构和外部中断处理流程如下图所示：



大致处理过程为：

- ü IO设备通过中断控制器(IO APIC或PIC)向CPU发送中断请求
- ü IO APIC将中断转发至目标CPU和Local APIC
- ü 目标APIC对该中断进行处理

在虚拟化环境中，VMM为Guest OS虚拟了一个与物理中断架构类似的虚拟中断架构，如下图所示：



每个VCPU对应一个虚拟Local APIC，用于接收中断，同时还模拟了虚拟IO APIC(或虚拟PIC)用于接收外设(虚拟设备)发出的中断请求并进行转发。虚拟Local APIC、虚拟IO APIC、虚拟PIC等都是VMM中的软件实体(对应于相应的数据结构)。主要处理流程如下：

- ü 当虚拟设备需要发送中断时，虚拟设备会调用虚拟IO APIC的接口发送中断；
- ü 而虚拟IO APIC根据中断请求，选出适合的虚拟Local APIC，并调用其接口发送中断请求；
- ü 虚拟Local APIC进一步利用CPU硬件虚拟化功能(Intel VT-x或AMD SVM)的事件注入机制，将中断注入到相应的VCPU；
- ü 当相应VCPU得到调度时，即处理相应的中断。

3.4 调度

在KVM虚拟化环境中，KVM虚拟机作为一个系统进程运行，因此虚拟机的调度，实际上就是利用了Linux自身的调度器完成。本文不做详述。

阅读(173) | 评论(0) | 转发(0) |

上一篇：[又一次内存分配失败\(关于overcommit_memory\)](#)
下一篇：[KVM基本原理及架构四-内存虚拟化](#)

0

相关热门文章

- 小编分享 挖掘Win7的XP模式里...
- linux 常见服务端口
- C语言 如何在一个整型左边补0...
- KVM基本原理及架构八-KVM内核...
- 【ROOTFS搭建】busybox的httpd...
- python无法爬取阿里巴巴的数据...
- KVM基本原理及架构七-KVM内核...
- xmanager 2.0 for linux配置
- linux-2.6.28 和linux-2.6.32....

KVM基本原理及架构六-KVM API...	什么是shell	linux su - username -c 命...
KVM基本原理及架构五-IO虚拟化...	linux socket的bug??	我不得不在这里问一下网站使用...

给主人留下些什么吧！~~

评论热议

请登录
后评论。
[登录](#) [注册](#)