基于显卡直接分配的虚拟机图形加速系统①

陈　诚　(复旦大学 软件工程系 上海 200433)

摘　要： 介绍了设计并实现的基于显卡直接分配的虚拟机图形加速系统 Gracias，它的思想是把显卡直接分配给某一台完全虚拟化的虚拟机使用。这样一来，虚拟机中显卡的设备驱动程序对于显卡的使用和访问操作就不会被虚拟机监控器所拦截，也不用通过软件模拟的方式来处理这些访问请求，而是直接交给真实的硬件去完成。这使得对图形处理要求很高的程序在虚拟机中可以获得比普通虚拟化方法高很多的性能提升。

关键词： 虚拟化技术；完全虚拟化；虚拟机监控器；直接分配显卡

Graphics Card Direct Assignment Based Virtual Machine Graphics Acceleration System

CHEN Cheng

(Department of Software Engineering, Fudan University, Shanghai 200433, China)

Abstract: This paper presents Gracias (Graphic Card Direct Assignment), which directly assign a graphics card to a certain virtual machine. Thus, access request from device driver of the graphic card in virtual machine will not cause any traps into the virtual machine monitor and the request will be handled directly by the hardware instead of software emulation. This makes the programs that have high demand on graphic processing give much better performance than with ordinary virtualization approaches.

Keywords: virtualization; full virtualization; virtual machine monitor; direct assignment; graphic card

本论文结构如下：首先介绍开源的虚拟机监控器 Xen[1]的整体架构，然后介绍我们基于 Xen 虚拟机监控器实现的虚拟机图形加速系统 Gracias(Graphic Card Direct Assignment)，最后进行系统性能及可靠性评测。

1　Xen虚拟机监控器

1.1 Xen 简介

Barham 等人开发的超级管理器(Hypervisor) 类型的虚拟机监控器。它既可以创建半虚拟化的虚拟机也可以创建完全虚拟化的虚拟机。

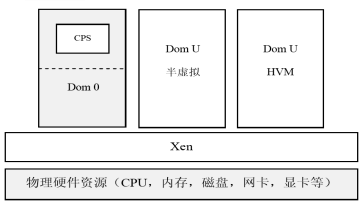

图 1　Xen 的整体架构

图 1 给出了 Xen 的整体架构。在系统中，由 Xen 管理着所有的物理硬件资源，包括 CPU，内存，磁盘，网卡，显卡等。Xen 作为系统中最高权限的软件，运行在 CPU 的 0 号特权级别(Ring 0)上。Xen 创建的每一个虚拟机，都叫做一个 Domain。如图 1，Xen 既可以创建半虚拟化的 Domain，也可以创建完全虚拟化的 Domain。

Xen 每次启动的时候都会创建一个半虚拟化的虚拟机，Domain 0。这个 Domain 在系统中具有特殊的意义。系统中每个 Domain 的设备虚拟化都依赖于 Domain 0，并且 Domain 0 还带有一个控制管理软件(Control Plane Software，CPS)。CPS 是运行在 Domain 0 用户态的一个程序，用户通过使用 CPS 可以创建或者销毁一台虚拟机，也可以对 Xen 进行配置。

除了 Domain 0 以外，Xen 上运行的所有的虚拟机都叫做 Domain U。而完全虚拟化的 Domain U 又叫做 HVM Domain(Hardware Virtual Machine

① 收稿时间:2009-11-22;收到修改稿时间:2009-12-25

Domain)，Xen 的完全虚拟化依赖于英特尔和 AMD 公司提供的硬件虚拟化辅助技术。

2 Gracias系统的设计与实现

2.1 整体架构

Xen 中 HVM Domain 的外设访问模型基于 QEMU[3]，在这种模型下，HVM Domain 中的驱动程序如果要访问某个 QEMU 模拟出的外设，对 CPU 的使用权需要经过多次虚拟机内部内核态和用户态的切换，两次虚拟机间切换以及两次虚拟机和虚拟机监控器间的切换。这种额外开销是十分庞大的，尤其对于显卡这种需要频繁高速地处理大量数据的设备。通过将显卡直接分配给 HVM Domain 使用，不仅减少了 HVM Domain 使用显卡时的大量不必要的开销，并且可以利用到显卡的图形加速功能。

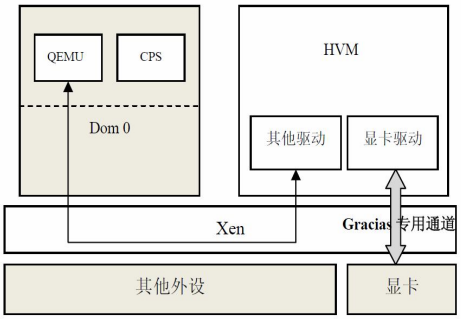

图 2 Gracias 系统的整体架构

图 2 显示了 Gracias 系统的整体架构。我们在 Xen 中建立了一个 Gracias 系统的专用通道，将 HVM Domain 的显卡驱动程序和 QEMU 解耦合，使其通过 Gracias 系统的专用通道直接访问物理显卡设备。由于我们在 HVM Domain 中安装的显卡驱动就是由该显卡厂商发布的专用设备驱动，因此这个驱动可以充分发挥显卡的全部特性，包括使用显卡的 GPU 和独立显存进行高速的图形运算和数据处理。而 HVM Domain 的其他设备驱动，仍然用 QEMU 模拟的方式对其设备进行访问。

下面我们从显卡设备注册，显卡资源直接访问，中断处理和 DMA 处理这四方面来详细介绍 Gracias 系统的实现。

2.2 显卡设备注册

2.2.1 将显卡注册到 QEMU 中

在 Xen 中，所有 HVM Domain 使用到的设备都是在 QEMU 中注册的。QEMU 中模拟了一条 PCI 总线（总线 0）以及注册在总线 0 上的磁盘控制器、CD 光盘驱动器、网卡以及其他必要的设备，同时它还模拟了一张 Cirrus 的 VGA 显卡，用于在 Domain 0 的窗口下显示 HVM Domain 的屏幕输出。

为了让 HVM Domain 使用真实的显卡而不是 Cirrus 显卡，我们将 Xen 中注册 Cirrus 显卡的逻辑替换成了注册真实显卡的逻辑。当在 QEMU 的 PCI 总线上注册了真实的显卡后，HVM Domain 启动的时候就能识别到真实显卡的存在了。

2.2.2 新 PCI 总线的注册

由于显卡是一种高速 PCI 设备，占用着系统大量的资源并且要求很大的数据交换量，因此显卡在系统中一般都独占一根 PCI 总线。为了让 HVM Domain 中的显卡驱动访问显卡能获得和真机上的驱动访问显卡一样的效果，我们将真机上连接显卡的 PCI 总线也直接分配给 HVM Domain。

图 3 显示了 Gracias 系统中的 PCI 结构。其中总线 0 和网卡、声卡、磁盘控制器等其他设备都是由 QEMU 负责模拟的。我们在 QEMU 中注册了一根新的总线。凡是在 QEMU 中注册过的设备，都会被 HVM Domain 识别到。这样，HVM Domain 中的显卡就出现在了和真机上相同的总线上，并且我们修改了 QEMU 注册 PCI 设备时给设备注册空闲的设备号和功能单元号的方式，让 QEMU 将显卡和 PCI 桥注册到和他们在真机上相同的设备号和功能单元号上。这样，在 HVM Domain 看来，显卡在 PCI 结构中的逻辑 BDF 号（即总线号、设备号、功能单元号）和真机上是完全一致的。

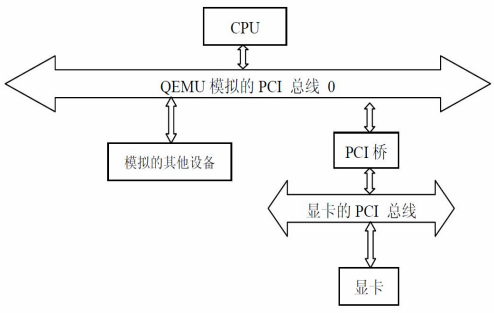

图 3 Gracias 系统的 PCI 结构

2.3 显卡资源直接访问

2.3.1 I/O 端口直接访问

与 iKernel[4]一样，我们通过设置 SVM[5]的 “I/O

权限位图”（I/O Permission Bitmap）来允许HVM Domain直接访问显卡设备的I/O端口。

“I/O权限位图”是AMD的SVM虚拟化技术提供给虚拟机监控器用于控制虚拟机I/O拦截的一种方式。每个虚拟机都对应有一张I/O权限位图，该位图是12千字节的连续物理内存空间。位图中的每个位代表一个单字节的I/O端口。如果某个I/O端口对应的那个位被置1，则虚拟机对该I/O端口的访问会被硬件拦截并且通知虚拟机监控器。如果某个I/O端口对应的那个位被置0，则虚拟机对该I/O端口的访问不会被硬件和虚拟机监控器拦截。每个虚拟机的I/O权限位图可以通过该虚拟机的VMCB中的IOPM_BASE_PA域找到。

我们将显卡会用到的标准VGA端口以及从显卡配置空间中读到的专用I/O端口相应的权限位都置为0，从而允许HVM Domain中的设备驱动对这些端口的直接访问。

2.3.2 VGA BIOS的使用

在系统启动的时候，系统BIOS通过调用显卡的VGA BIOS来对显卡进行初始化。原来HVM Domain启动时会调用Cirrus显卡的VGA BIOS来初始化显卡，但是它不能用来初始化真实的显卡。所以我们从真机的显卡ROM中将它的VGA BIOS读取出来，在HVM Domain启动前将这段VGA BIOS代码加载到原来存放Cirrus显卡VGA BIOS的机器物理内存地址上，这样HVM Domain就能用正确的VGA BIOS来初始化显卡了。

2.4 中断处理

在Xen虚拟化的环境中，一切来自外设的中断都是由Xen接受的。当虚拟机运行时，CPU收到一个中断到达的信号后，硬件会产生一个VMExit进入到root状态中，将CPU使用权交给Xen。

在Gracias系统中，如果Xen收到了来自设备的中断，并且判断出该中断向量属于显卡，就会发送一个EOI（End of Interrupt）给Local APIC并且将这个中断注入到HVM Domain中。

在VMCB（Virtual Machine Control Block）结构中有一个子结构VINTR（Virtual Maskable Interrupt），VINTR结构中有一个域vector用来保存中断的向量号。当Gracias系统确定中断来自显卡设备时，就将该vector域设置成该中断向量号，这样进入HVM Domain后，操作系统就会调用该向量号所指向的中断处理函数，而该中断处理函数正是显卡设备驱动注册的。这样，显卡发出的中断就能够被HVM Domain中的显卡设备驱动正确的处理了。

2.5 DMA处理

在Gracias系统的设备访问模型下，显卡驱动对DMA的使用会带来两个问题。

第一个问题是HVM Domain的物理地址和真机物理地址的不一致性。影子页表机制只是解决了CPU访问内存时HVM Domain物理地址空间和真机物理地址空间的不一致问题。对于显卡，显卡驱动要求其进行DMA的地址范围是根据虚拟机内的物理地址确定的，这和显卡所见的真机物理地址是不一样的。

第二个问题是安全问题。HVM Domain中的恶意驱动程序可以通过DMA的方式攻击Xen。

AMD公司SVM IOMMU[6]和英特尔公司的VT-d[7]技术解决了DMA重映射的问题。

有了这种重映射机制，我们可以给显卡设备设置一张DMA重映射表，当显卡要对某个物理地址进行DMA操作时，IOMMU的硬件会自动查找DMA重映射表，帮助显卡进行虚拟机内物理地址到真机物理地址的转换。

Gracias系统保证了P2M表的映射关系能够如实地反映到显卡的DMA重映射表上。当硬件由于在DMA重映射表中找不到地址映射关系产生一个重映射错误的时候，Gracias系统会查找P2M表，找到该地址对应的虚拟机内物理地址，并且将DMA重映射表相应的表项填好。如果驱动试图操纵显卡用DMA的方式访问Xen保护的物理地址，也会被IOMMU硬件拦截到，交给Gracias系统进行适当的处理，最后阻挡恶意攻击的发生。

3 Gracias系统性能及可靠性评测

3.1 系统配置

Gracias系统的测试机使用AMD Atholon? 64 X2 Dual Core 6000+处理器，华硕M2N-SLI型号主板，4GB的DDR2内存，300GB硬盘。配备两张ATI Radeon HD 2600 XT型号的显卡，每张显卡有256MB的显存。

Gracias系统基于Xen 3.1.0版本，Domain 0操作系统是Fedora Core 6，内核为Linux 2.6.18

版本，HVM Domain 安装 Windows Vista™ Ultimate 操作系统。每个 HVM Domain 配有 1GB 的内存和 50GB 硬盘空间。

3.2 3DMark® 06 评测

我们使用 3DMark® 06 1.1.0 专业版[8]测试 HVM Domain 的图像处理能力。

当我们尝试在未直接分配显卡的 HVM Domain 中运行 3DMark 06 时，由于系统显卡（QEMU 模拟的 Cirrus 显卡）没有 3D 图形硬件加速功能而无法打开测试程序。

我们在以下 3 个场景下用 3DMark® 06 分别测试了系统的图像处理能力。

场景一：真机上的 Windows Vista™ Ultimate 操作系统，非虚拟化环境。

场景二：在只开启一个 HVM Domain 的环境下，将显卡直接分配给这个 HVM Domain。

场景三：同时开启两个 HVM Domain，将两块显卡分别分配给两个 HVM Domain。测试两个 HVM Domain 图像处理能力的平均值。

表 1 列出了这 3 种不同场景下的 3DMark® 06 得分。其中 SM 指 Shader Model，HDR 指 High Dynamic Range，都是目前流行的硬件图形加速技术标准。

从表中对比场景一和场景二可以看到，直接分配了显卡的 HVM Domain 的图形处理能力是真实物理机图形处理能力的 92%左右。而对比场景二和场景三

表 1 3DMark® 06 得分

项目	场景一	场景二	场景三
3D Mark 得分	4775	4383	4216
SM2.0	1664	1535	1483
HDR/SM3.0	2068	1884	1807

可以看到，同时运行两个 HVM Domain 的话，图形处理能力相对只运行一个 HVM Domain 时仅有 4%左右的下降。在运行 3DMark® 06 的 CG 动画时，场景二与场景三均不会有明显的不流畅和掉帧的感觉。

3.3 安全性及可靠性评估

iKernel 一文中提出，将设备驱动对设备的访问限制在一个虚拟机内，可以十分有效地防止设备驱动错误的传播，并且可以防止内核态恶意的设备驱动攻击操作系统和用户程序。

Gracias 系统将显卡直接分配给 HVM Domain 从而大大提高其图形处理能力的同时，限制了 HVM Domain 对真机上资源的访问仅局限在显卡使用的那些 I/O 端口、内存映射 I/O、显存和 VGA 帧缓冲区。显卡的中断和 DMA 也完全在 Gracias 系统的掌控之下完成。因此将显卡直接分配给 HVM Domain 并不会降低整个系统的安全性。

此外，由于 Gracias 系统的显卡驱动是运行在虚拟机中的，如果显卡驱动存在漏洞或者缺陷，其影响至多导致虚拟机崩溃，而不会影响到系统的其他任何部分。因此 Gracias 系统对显卡的直接分配不会对系统的整体可靠性造成任何影响。

参考文献

1 Barham P, Dragovic B, Fraser K, Hand S, Harris T, Ho A, Neugebauer R, Pratt I, Warfield A. Xen and the art of virtualization. SOSP '03: Proc. of the nineteenth ACM Symposium on Operating Systems Principles. New York, NY, USA: ACM, 2003. 164－177.

2 Adams K, Agesen O. A comparison of software and hardware techniques for x86 virtualization. ASPLOS-XII: Proc. of the 12th International Conference on Architectural Support for Programming Languages and Operating Systems. New York, NY, USA: ACM, 2006.2－13.

3 Bellard F. Qemu, a fast and portable dynamic translator. ATEC'05: Proc. of the USENIX Annual Technical Conference 2005 on USENIX Annual Technical Conference. Berkeley, CA, USA: USENIX Association, 2005.41.

4 Tan L, Tan L, Chan EM, Farivar R, Mallick N, Carlyle JC, David FM, Campbell RH. ikernel: Isolating buggy and malicious device drivers using hardware virtualization support ikernel: Isolating buggy and malicious device drivers using hardware virtualization support. in Dependable, Autonomic and Secure Computing, 2007. DASC 2007.Third IEEE International Symposium on, 2007.134－144.

5 AMD Virtualization Technology, AMD.[2009-9-3].

(下转第 33 页)

(上接第 9 页)

http://www.amd.com/us/products/technologies/virtualization/Pages/virtualization.aspx

6 I/O Virtualization and AMD’s IOMMU, AMD.[2009-9-3].http://developer.amd.com/documentation/articles/pages/892006101.aspx

7 Intel Virtualization Technology, Intel.[2009-9-3]. http://www.intel.com/technology/itj/2006/v10i3/2-io/7-conclusion.htm

8 3D Mark 2006, Futuremark.[2009-10-13]. http://www.futuremark.com/benchmarks/3dmark06/download/