基于共享内存的Xen虚拟机间通信的研究

朱团结,艾丽蓉
(西北工业大学 计算机学院,陕西 西安 710072)

摘 要:当虚拟机技术应用在服务器整合等领域时,虚拟机之间的通信会非常频繁,虚拟机本身的通信机制将成为瓶颈。目前,在Xen中不同的虚拟机间进行通信时,不仅通信路径长,而且虚拟机间的切换会造成很大的性能开销。在深入研究Xen虚拟机通信机制的基础上,提出了一种基于共享内存的通信方法,用于提高同一台物理机器上不同虚拟机之间通信的性能。实验结果表明,该方法极大地增加了虚拟机之间的通信带宽,并且有效平衡了各个虚拟机的CPU利用率。

关键词:虚拟机;通信;Xen;共享内存;带宽
中图分类号:TP31 **文献标识码**:A **文章编号**:1673-629X(2011)07-0005-04

Research on Xen Inter Domain Communication Based on Shared Memory

ZHU Tuan-jie, AI Li-rong
(Dept. of Computer, Northwestern Polytechnical University, Xi'an 710072, China)

Abstract: When the virtual machine technique is applied to many fields such as the server integration, the communication inter virtual machines will be very complex and frequent, and so the communication mechanism of the virtual machines themselves will become a bottleneck. Currently, when one guest virtual machine in Xen wants to communicate with another on the same physical machine, it must extend the data transfer path and degrade the inter-domain communication performance. On the basis of an intensive study of the communication of the Xen virtual machines, it puts forward a communicating method based on the shared memory, which is used to improve the performance of the communication between different virtual machines on one physical machine. The experimental result shows that this method has greatly increased the communication bandwidth between virtual machines, and effectively balanced the CPU utilization of every virtual machine as well.

Key words: virtual machine; communication; Xen; shared memory; bandwidth

0 引 言

近年来,随着虚拟化技术的不断发展,逐渐地出现了大量性能优异、稳定的虚拟化技术,其中Xen虚拟机就是典型的代表。Xen虚拟机具有高效性、开源性等特点,最初它只支持半虚拟化技术,硬件虚拟技术出现后,使Xen能够支持完全虚拟化技术。虚拟机技术能够有效屏蔽硬件平台的异构性和动态性,使得硬件资源得到最大限度的共享和复用,因此,应用范围极其广泛。但是,虚拟机技术的使用也遇到了一些问题,比如,服务器整合、集群运算等技术对虚拟机之间的通信性能要求较高,这就使得同一台物理机器上各虚拟机间通信出现瓶颈[1]。文中针对这一问题,提出一种基于共享内存的通信方法。

收稿日期:2010-12-13;**修回日期**:2011-04-06
基金项目:国家自然科学基金(60273087);西北工业大学基础研究基金(W018101);陕西省自然科学基金(SJ08F25)
作者简介:朱团结(1984-),男,硕士研究生,研究方向为网络与信息安全;艾丽蓉,博士,副教授,研究方向为人工智能。

1 Xen体系结构及通信机制

文中的主要内容是研究如何改进Xen虚拟机的通信机制,实现基于共享内存的通信方法。因此,首先应该对Xen虚拟机的框架结构和其本身技术自带的通信机制进行全面和深刻的理解和掌握,然后,才能在此基础上实现基于共享内存的通信机制。

1.1 Xen的体系结构

一个Xen虚拟化环境由以下部件构成:虚拟机监控器VMM、特权虚拟机和客户虚拟机。虚拟机监控器VMM直接安装在已有的硬件环境上,它的主要任务是实现虚拟机技术的各种底层机制,如虚拟化CPU、虚拟化设备、虚拟化内存和虚拟化网络等机制都是在HMM中实现的,同时,虚拟机监控器HMM还对安装在其上的虚拟机提供一些基本的控制[2]。安装完成虚拟机监控器后,在它上面再安装各种虚拟机,主要包括特权虚

拟机和客户虚拟机。特权虚拟机是最先被创建的虚拟机,同时,它也作为虚拟机监控器的扩充而存在,它直接拥有系统的硬件输入和输出设备的控制权,提供对这些I/O设备响应的驱动程序,并且支持在用户界面上对客户虚拟机的管理功能。其中,特权虚拟机必须在其它虚拟机启动之前启动,它主要包含两个驱动:Network Backend Driver和Block Backend Driver,分别负责处理来自客户虚拟机的网络请求和本地磁盘请求。Network Backend Driver直接和本地网络硬件进行通信以响应所有来自客户虚拟机的网络请求。Block Backend Driver和本地的存储设备进行通信以处理来自客户虚拟机的读写请求。这两个驱动也是完成虚拟机间通信的必要条件[3]。特权虚拟机的设备模型和驱动程序可以为虚拟机之间的网络通信提供最基础的技术支持,具体的实现机制主要有虚拟网络、事件通道等。

另外,Xen支持Intel VT-X技术,该技术有一个关键的VMX操作和一个数据结构VMCS。VMX可以在虚拟机监控器和客户虚拟机下执行,VMCS则定义了特权虚拟机和客户虚拟机的转换以及在转换过程中虚拟机监控器和上层各种虚拟机的状态信息等。Intel VT-X技术可以通过数据结构VMCS和一组虚拟机特有的扩展指令对虚拟机的进入操作(VMentry)和退出操作(VMexit)进行控制,以实现虚拟机间的切换,在VMentry操作和VMexit操作的实现过程中,虚拟机监控器和其上的各种虚拟机的状态都保存在数据结构VMCS中。也正是这种切换机制使得客户操作系统可以不修改内核就能够直接安装在Xen虚拟机中,同时,这种机制能够快捷方便地实现虚拟机监控器HMM对其上安装的各虚拟机的管理和控制[4]。这样,既保存了虚拟机技术良好的特性,又使其具有通用性,为虚拟机技术的广泛应用奠定了基础。

1.2 Xen虚拟机通信机制

在以上对Xen虚拟机体系结构了解的基础上,知道虚拟机技术最主要的特点就是对硬件环境的共享和复用。实现虚拟机通信的最主要的方式就是虚拟网络,即通过使用虚拟网卡实现网络通信。在日常的使用中,网络通信都是通过网卡与网卡之间的连接实现的,但是,在虚拟机技术中硬件网卡数量有限,不可能为每一个虚拟机都加入一个相应的网卡,只能将网卡虚拟化,各个虚拟机共享使用。其中,同一物理计算机上的虚拟机间的通信可以直接通过各自的虚拟网卡实现,而虚拟机与其他物理计算机或者其他物理计算机上的虚拟机之间的通信则用机器的物理网卡实现硬件与外部网络的通信。使用虚拟网卡技术以后,就可以把虚拟机当成整个网络中的一台独立机器,它能够与主机和主机上的其他虚拟机以及其他真实的物理机器之间通过网络进行数据传输。这种通信对于用户来说是透明的。虚拟网卡能够像物理网卡那样对数据包进行发送和接收,它们的首发原理基本相同,主要区别在于,虚拟网卡主要借助于tun/tap驱动来实现数据传输的功能,达到网络传输的目的。因为虚拟网卡是多个虚拟机共享复用的,在实际使用过程中它需要通过一系列的控制寄存器保存状态信息来实现对虚拟网卡的切换和控制。这些不同,在Xen虚拟机技术中体现为:在虚拟机进行I/O请求和响应的时候,会自动触发退出VMexit操作和进入VMentry操作,以引起客户虚拟机和虚拟机监控器之间的切换,完成需要通信的虚拟机对虚拟网卡的控制。此外,在Xen虚拟机中还可以通过事件通道来实现同一物理机器上虚拟机间的简单通信,比如传递一些基本的对虚拟硬件的控制信息等,它和中断机制非常相似。事件通道是虚拟机技术用于虚拟机和虚拟机之间、虚拟机和虚拟机监控器HMM之间的一种异步事件通知机制,在虚拟网络中,一些虚拟机监控器HMM对虚拟机事件的通知就是通过事件通道完成的,比如当客户虚拟机向虚拟网卡发送请求时,虚拟机监控器就会通过事件通道向特权虚拟机的设备模型传递请求。虚拟机的中断系统包括:物理中断、虚拟处理器间中断、虚拟中断,在Xen虚拟机中每一个中断都是一个事件,它们分别对应一条事件通道,当中断发生时,通过对应的事件通道传输消息[5]。事件通道这一机制在虚拟机通信中极为重要,它不仅是中断和控制信息的传输通道,也是实现虚拟网卡通信的关键技术基础,文中提出的通信方式,也需要事件通道的技术支持。

在Xen虚拟技术的通信机制中,最重要的环节就是通过VMCS数据结构对VMX虚拟机的进入操作和退出操作的控制。可以通过对虚拟机的VMexit操作和VMentry操作完成虚拟机读写虚拟网卡的请求和对请求的响应以及其他对虚拟机各种操作。比如,当虚拟机要从网卡中读取数据包时,其I/O请求就会触发VMexit操作,使得系统将处理器的控制权交给虚拟机监控器HMM,与此同时,虚拟机监控器会将提出I/O请求的各种信息写入数据结构VMCS中,然后,虚拟机监控器会通过事件通道将请求发送给特权虚拟机的设备模型进行相应的处理,设备模型的主要工作是在识别外设访问的类型后调用相应的I/O处理函数。如果最后虚拟机发出的I/O请求得到了相应的处理,虚拟机监控器将会通过执行VMentry操作通知虚拟机接收数据包[6]。这样一次网络I/O请求至此完成,在整个过程中虚拟机间的切换比较频繁。

相关的研究结果表明[7],当Xen虚拟机系统处于

运行状态时,网络传输方面的开销所占的比重是非常巨大的,直接影响着整个系统的性能。在特权虚拟机中网络开销大约占到了31%左右,在客户虚拟机中几乎占到了41%左右。造成开销的主要原因是TCP/IP协议栈开销和虚拟机间切换等。

2　基于共享内存的通信模型

如图1所示,通信模型由两部分组成:特权虚拟机代理和客户虚拟机代理。

特权虚拟机代理负责发现一台物理机器上的所有客户虚拟机,并将虚拟机的ID、MAC地址等信息存放在一个表中。客户虚拟机代理查询与其通信的目标虚拟机是否位于同一台物理机器上,如果在同一台物理机器上,则建立共享内存,绕过TCP/IP协议进行通信。通信过程中,共享内存上存在两个数据通道,控制信息通道由Xen自身机制完成。

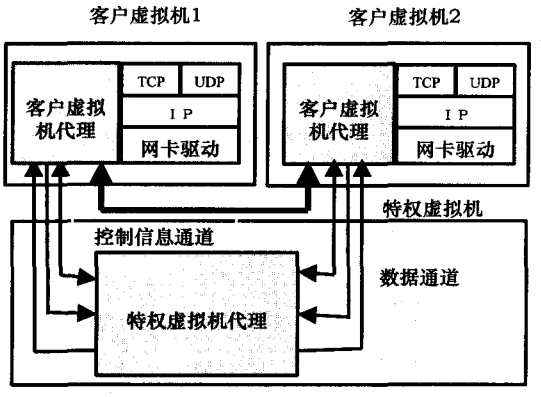

图1　虚拟机间通信模型

2.1　特权虚拟机代理

在Xen虚拟机中,Xenstore机制[8]是不同虚拟机间共享的存储区,通常用于存储配置和状态信息,不能用于大量数据的通信。该机制以键-值的形式存储,它主要包含了每个虚拟机的相关的值和存储库信息以及许可证等内容。Xenstore机制可以用于实现Xen虚拟机中的应用程序和驱动之间相互通信并将配置信息存储下来,具体的实现过程为:应用程序可以通过写数据库中的键-值信息的方式来使用Xenstore机制配置驱动,驱动则会在相关的键-值上设置监视,当发现键-值发生改变时,便会做出相应的反应。

在基于共享内存的通信模型中,特权虚拟机代理的主要工作是:首先利用Xenstore这一机制获取一台物理机器上的所有信息表格,以便客户虚拟机发出查询请求时,可以通过写Xenstore来响应相应的查询请求。Xenstore机制可以通过调用xenbus_directory函数来获取同一台物理机器上所有虚拟机的信息,然后,只需要读取Xenstore的路径可获取所有客户虚拟机的ID。另外,可使用xenbus_read和xenbus_printf函数读取某一个键值和写入某一个键值[9]。

2.2　客户虚拟机代理

在实际应用中,同一台物理机器上的多个虚拟机之间经常需要交互,甚至在一些领域中,这种交互极其频繁。通常,解决这类通信问题的最好方式是共享存储空间,因此,Xen虚拟机自身提供了一种共享内存页的机制,各个虚拟机可以在共享内存页的基础上实现一些共享策略,用于满足不同虚拟机之间通信的需求。Xen虚拟机的共享内存是以页为基本单位的,而这些页都可以通过一个整数来简单的表示,这一系列的整数指向了Grant Table中的相应的入口,也可以称之为Grant Table的索引。在Xen虚拟机中,每个虚拟机都有一个相对应的Grant Table,它是一个和Xen共享的数据结构。客户虚拟机代理就是在共享内存的基础上设计完成的。在该通信模型中,绝大部分工作是由客户虚拟机代理完成的。它的主要任是:

1)获取目标机器的MAC地址;

2)向特权虚拟机上的特权虚拟机代理发送查询请求,特权虚拟机代理查询自己维护的表格,然后告知客户虚拟机代理目标机器的ID;

3)需要通信的目标机是否在同一台物理机器上;

4)通过Xenstore传递必要的控制信息;

5)使用Xen的Grant Table机制建立共享内存[10];

6)在Xenstore中新建一个键值,其中包含共享内存的各项信息;

7)通过共享内存进行数据传递;

8)若通信结束,释放共享内存空间。

在共享内存建立前,需要传递一些必要的控制信息,这些信息都可以通过对Xenstore机制的使用来传递。另外,由于共享内存是多个内存页组成的,对每个内存页都需要做相应的信息传递操作,如果这些信息全部通过Xenstore机制来传递,系统将会对Xenstore进行频繁的读写,降低通信性能。因此,首先建立一个描述符页,该页内用一个数组来记录所有的信息,只要Xenstore将该描述符页映射过去,那么就可以获得所有的共享页。

2.3　通信的实现

该通信模型的两种连接方式是:面向连接的Stream Connection和面向无连接的Datagram Connection,它们分别相当于TCP和UDP通信。

Stream Connection的工作流程如图2所示,服务器首先等待客户端的连接申请,建立连接后,它们之间便可以进行通信。然而,Stream Connection只能有两个终端,不能在客户端里再创建另外一个连接,这也是它与TCP协议的最大区别。

Datagram Connection 的工作流程如图3所示，服务器一直等待客户端发送数据并接收，与 Stream Connection 相比，该连接的数据传输可靠性较低，但是数据传输效率更高。

在共享内存建立以后，可以根据通信的实际情况，选择一种通信方式。当通信量较大且可靠性要求不高时可以选择 Datagram Connection 连接方式，对可靠性要求高的，可选择 Stream Connection 连接方式。

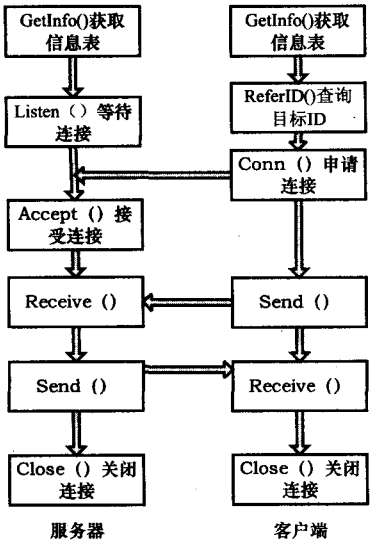

图2 Stream Connection

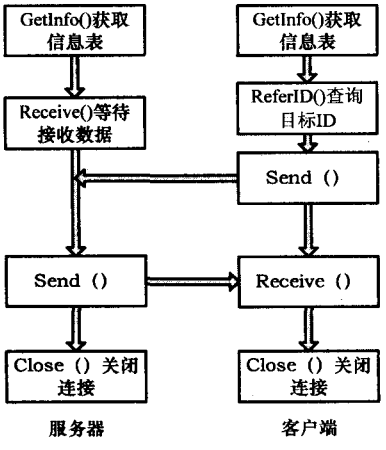

图3 Datagram Connection

3 实验及结果

实验使用 HP DL140G3×5130（支持 VT 技术的5130CPU），并安装 CentOS5.2 系统。创建2个 CentOS5.2 虚拟机 VM1 和 VM2，虚拟内存为700M。实验从带宽和 CPU 利用率两个方面对基于共享内存的通信方式进行评估[11]。

为了测试带宽，实现了一个可以改变缓冲区大小的发送函数，VM1 将300M 的文件在内核中打开，然后发送到 VM2。表1中实验结果表明基于共享内存的通信方式可以将带宽提高到2000Mbps，相关研究结果表明[12]，Xen 虚拟机的通信机制的带宽是100Mbs 左右，这就极大地提高了通信的带宽。

在传输数据的同时，借助 Xen 的 xentop -b - i 2 | awk '{print \$4}' 命令提取特权虚拟机和客户虚拟机的 CPU 利用率信息，结果如图4所示，基于共享内存的通信模型平衡了虚拟机间的 CPU 利用率，特权虚拟机的 CPU 利用率大幅度降低，使它能够为更多的虚拟机提供服务。

表1 共享内存通信方式带宽

缓冲区大小	带宽(Mbps)
256B	278.485
1K	624.092
16K	2365.639
64K	1893.218
128K	2053.603

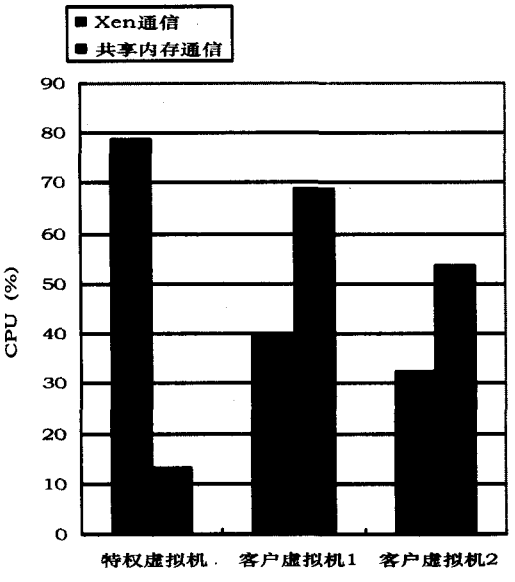

图4 CPU 利用率对比

4 结束语

文中在深入研究 Xen 虚拟机通信机制的基础上，提出一种基于共享内存的 Xen 虚拟机间通信模型。通过实验证明这种方法有效提高了虚拟机间通信的效率，但是该模型还没有保证通信安全的机制，下一步拟采取一些措施提高模型的安全性。

参考文献：

[1] 怀进鹏，李 沁，胡春明. 基于虚拟机的虚拟计算环境研究与设计[J]. 软件学报，2007，18(8)：2016-2026.

[2] 黄良良，韩 军，汪伦伟. 基于 Xen 硬件虚拟机的安全通信机制研究[J]. 计算机安全，2010(3)：30-31.

[3] Fraser K, Hand S, Neugebauer R, et al. Safe hardware access with the Xen virtual machine monitor[C]//Proceedings of the 1st Workshop on Operating System and Architectural Support for the on Demand IT Infrastructure (OASIS). Bos-

（下转第12页）

参与人员:<参与者>

应急资源:<应急物资>

处置过程:<处置指令>

本框架体系的总体结构如图1所示。

4 结束语

应急案例中涉及到的知识内容较广,框架方法适合表达静态结构的知识,因此其主要的应用场合有两类:

·用于描述案例中的人员、事件、资源、指令等四项以静态特征为主的内容;

·用于在案例库建设初期,对于搜集到的大量案例(由于是事后收集的,因此通常缺乏详细的处置过程信息)进行描述,用作案例库中的初始数据。

理想化的案例描述应当是在事件发生过程中同时收集有关信息,转化为框架表达方式存入案例库,但是对于案例的动态演进过程则需要采用别的更合适的知识表达方法。

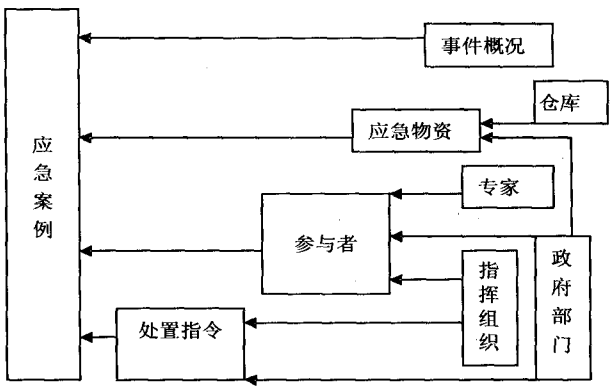

图1 应急案例的框架体系总体结构

参考文献:

[1] 张英菊,仲秋雁,叶 鑫,等. CBR的应急案例通用表示与存储模式[J]. 计算机工程,2009,35(17):28-30.

[2] 王文俊,杨 鹏,董存祥. 应急案例本体模型的研究及应用[J]. 计算机应用,2009(5):1437-1440.

[3] 蒋 鹏. 基于本体构建的应急案例库的研究[J]. 高等职业教育:天津职业大学学报,2009,18(2):86-88.

[4] 谢红薇,李建伟. 基于本体的案例推理模型研究[J]. 计算机应用研究,2009,26(4):1422-1424.

[5] 李 华,赵道致,范 文,等. 基于SUMO的应急预案本体[J]. 情报学报,2009(3):331-338.

[6] 廖振良,刘宴辉,徐祖信. 基于案例推理的突发性环境污染事件应急预案系统[J]. 环境污染与防治,2009,31(1):86-89.

[7] 王天成. 基于案例推理的应急预案管理研究[J]. 现代计算机:下半月版,2008(7):40-43.

[8] 郭泳亨,卢兴华,刘 云. 基于案例库的应急决策支持系统研究[J]. 微计算机信息,2006(22):148-150.

[9] Malizia A,Onorati T,Diaz P ,et al. SEMA4A: An ontology for emergency notification systems accessibility[J]. Expert Systems with Applications,2010,37(4):3380-3391.

[10] Levy J K,Taji K. Group decision support for hazards planning and emergency management: A Group Analytic Network Process (GANP) approach[J]. Mathematical and Computer Modelling,2007,46(7-8):906-917.

[11] Sang Tae Chung,Kwang Il Kim. Case studies of chemical incidents and emergency information service in Korea[J]. Journal of Loss Prevention in the Process Industries,2009,22(4):361-366.

[12] 张旭凤. 应急物资分类体系及采购战略分析[J]. 中国市场,2007(32):110-111.

(上接第8页)

ton,USA:[s. n.],2004:1-10.

[4] 顾晓峰,王 健. 基于Intel VT-x的XEN全虚拟化实现[J]. 计算机技术与发展,2009,19(9):242-245.

[5] Barham P, Dragovic B, Fraser K, et al. Xen and the Art of Virtualization[C]//The nineteenth ACM symposium on operating systems principles. New York, NY, USA:[s. n.], 2003:164-177.

[6] Cho Y C, Jeon J W. Sharing Data Between Process Running on Different Domains on Para-virutalized Xen[C]//International Conference on Control. Automation and Systems. [s. l.]:[s. n.],2007:1255-1260.

[7] Menon A, Santos R J, Turner Y, et al. Diagnosing Performance Overheads in the Xen Virtual Machine Environment[C]//Proc of the 1st ACM/USENIX International Conference on Virtual Execution Environments. Chicago, USA:[s. n.], 2005:13-23.

[8] Clark C,Fraser K,Hand S,et al. Live Migration of Virtual Machines[C]//Proceedings of the 2nd ACM/USENIX Symposium on Networked Systems Design and Implementation (NSDI). Boston, USA:[s. n.], 2005:273-286.

[9] 张 建. Xen虚拟机间通信优化研究与实现[D]. 上海:上海交通大学,2008.

[10] 赖宗澐. Xen虚拟机存储系统优化[D]. 杭州:浙江大学,2007.

[11] 王太成,蔡 勇. 利用虚拟机技术完成复杂网络实验[J]. 计算机技术与发展,2009,19(4):246-249.

[12] Zhang Xiaolan, Mcintosh S, Rohatgi P, et al. XenSocket:A High-Throughput Interdomain Transport for Virtual Machines [C]//Proc of Middleware. Berlin:Springer, 2007:184-203.