Xen的虚拟机网络优化研究

孟江涛，卢显良，董贵山

(电子科技大学计算机科学与工程学院 成都 610054)

【摘要】针对一类流行的IP网络应用，提出了一个性能优化的虚拟机网络原型。多个虚拟机运行在高性能虚拟机监控器Xen上，通过它复用了其下的一台物理宿主机，Xen创建和管理这些虚拟机。优化原型的核心是一个新的虚拟网卡，所有的虚拟机通过它被互连成一个网络，用于虚拟机间的通信。与Xen的标准相应模型相比，实验和评估表明该原型改善了虚拟机间的通信性能，减少了约42%的用户请求响应时间。

关 键 词 复用; 网络设计; 网络性能; 虚拟机监控器; 虚拟机; Xen

中图分类号 TP316 **文献标识码** A doi:10.3969/j.issn.1001-0548.2010.01.024

Optimizing Communication Network for Virtual Machines Based on Xen

MENG Jiang-tao, LU Xian-liang, and DONG Gui-shan

(School of Computer Science and Engineering, University of Electronic Science and Technology of China Chengdu 610054)

Abstract For a class of popular IP network applications, a prototype of performance-optimized communication network for virtual machines is proposed. A few virtual machines run on the top of Xen, which is high-performance virtual machine monitor and multiplexes the underlying physical host. Xen creates and manages these virtual machines. A new virtual network card is the core of the prototype. All virtual machines are interconnected by this card into a network. Compared with default model of Xen, the prototype can improve the communication performance and reduce the response time for request by 42%.

Key words multiplexing; network design; network performance; virtual machine monitor; virtual machine; Xen

虚拟机监控器(VMM)虚拟计算平台的硬件资源，以支持多个虚拟机(VM)的同时运行。每个虚拟机独立运行一个操作系统，运行于虚拟机内的操作系统被称为客户操作系统(GOS)，虚拟机监控器为这些操作系统提供安全和高度的隔离[1]。

Xen 是运行于 Intel x86 上的 VMM，它支持多个 GOS 所未有的性能和隔离性同时运行[1]，是遵循 GNU 许可的开源软件[2]。当前，运用 Xen 支持多个虚拟机，并且在每个虚拟机上各自运行单独的操作系统，复用计算平台的研究正逐渐成为国内外学者研究的热点[3-5]。

Xen 能为流行的 3 层架构互联网应用[6]提供复用计算平台。3 层架构互联网应用包括：(1) 前端是 HTTP 服务器，负责处理用户输入输出；(2) 中间是应用服务器，实现应用的核心功能；(3) 后端是数据库服务器[7]，存储用户数据。3 层架构互联网的一个突出特点是用户仅与 HTTP 服务器交互，不与另外两个服务器交互。换言之，HTTP 服务器是访问另外两个服务的单点入口或应用网关。

针对 3 层架构互联网应用，本文提出了在同一台宿主机上基于 Xen 的多个虚拟机间的通信网络优化设计方案：在同一台宿主机上，Xen 创建和管理多个虚拟机，3 层架构互联网应用中的每一层服务单独运行于一个虚拟机中，为每个虚拟机配置更高性能的虚拟网卡，所有虚拟网卡被互连成一个虚拟机网络。

1 Xen的标准虚拟机网络模型

Xen 启动时创建的第一个虚拟机被称为控制虚拟机，简记为 Domain0。Domain0 被赋予特权，负责创建、管理其他虚拟机，被称为用户虚拟机(UVM)。

控制环(control ring)是 Domain0 与 UVM 的通信缓冲区。Domain0 通过控制环分配一个物理内存页，

收稿日期：2008 – 07 – 06； 修回日期：2009 – 05 – 18
基金项目：信息产业部电子发展基金(信部运[2002546])
作者简介：孟江涛(1971–)，男，博士生，主要从事操作系统结构、计算机网络等方面的研究.

使任意两个 UVM 通过各自的虚拟地址访问该页，从而共享该物理内存页。控制环是虚拟机间共享内存通信环和事件通道机制的基础。

定义 1　假定 Xen 正运行 N 个虚拟机，定义虚拟机间共享内存通信环 SMR 为一个二元组：

$$SMR(VM_i,VM_j)=\langle PF_{ij}, PF_{ji}\rangle \quad i, j=1,2,\cdots,N;\ i\neq j$$

式中　PF_{ij} 是一个物理内存页，VM_i 向 PF_{ij} 写消息，VM_j 从 PF_{ij} 读消息；PF_{ji} 是另一个物理内存页，VM_j 向 PF_{ji} 写消息，VM_i 从 PF_{ji} 读消息。

定义 2　假定 Xen 正运行 N 个虚拟机，定义虚拟机间事件通道 IEC 为一个二元组：

$$IEC(VM_i,VM_j)=\langle ECVM_i,ECVM_j\rangle$$

式中　$i, j=1,2,\cdots,N$；$i\neq j$；$ECVM_i$ 是 VM_i 的一个事件通道；$ECVM_j$ 是 VM_j 的一个事件通道。IEC 具有以下性质：

(1) VM_i 置 VM_j 的 $ECVM_j$ 位，这是 VM_i 向 VM_j 发送的异步通知。反之亦然。

(2) VM_i 把 $ECVM_i$ 映射成一个中断源，称为虚拟中断源，它的置位将触发 VM_i 上的中断。同理，$ECVM_j$ 是 VM_j 的一个虚拟中断源。

文献[2]描述了 Xen 的设备 I/O 模型，标准的虚拟网卡遵循该模型。控制虚拟机 Domain0 把创建的 UVM 分为两类，一类有直接访问物理网卡的特权，称为设备虚拟机，简记为 DVM.Domain0，启动 DVM 时，其 GOS 装载物理网卡驱动程序；另一类无此特权，称为非特权虚拟机，简记为 UPVM.Domain0，启动 UPVM 时，先根据其启动配置文件创建虚拟网卡，再由其 GOS 装载相应的驱动程序。在 Xen 的支持下，UPVM 的虚拟网卡及其在 GOS 中的驱动程序完全由软件实现。

Xen 把虚拟网卡分成前端和后端两部分，前端在 UPVM 中，后端在 DVM 中，前端和后端代理了 UPVM 中虚拟网卡驱动程序和 DVM 中物理网卡驱动程序之间的交互。每一个 UPVM 所接收或发送的包，都要通过共享内存环在前端和后端间转移，也要通过 DVM 中的网桥或交换设备在后端和物理网卡间转移。所以，同一台宿主机上的任何虚拟机间的网络通信都不得不通过 DVM，DVM 是虚拟机间网络通信的瓶颈。

Xen 通过虚拟机间事件通道 IEC 在虚拟机间触发异步通知，通过虚拟机间共享内存通信环 SMR 在虚拟机间传递消息。为加快 I/O，Xen 还运用零拷贝页重映射机制在前端和后端之间交换数据页。

UPVM 的 GOS 负责虚拟网卡的初始化，将虚拟网卡初始化为标准的以太网设备(IEEE 802.3)，其 MTU 值(最大传输单元)为 1 500 B。

2　优化的虚拟机网络模型

本文仅研究在同一台宿主机上被 Xen 所虚拟的多个虚拟机间的网络优化问题，优化模型体系结构如图 1 所示。

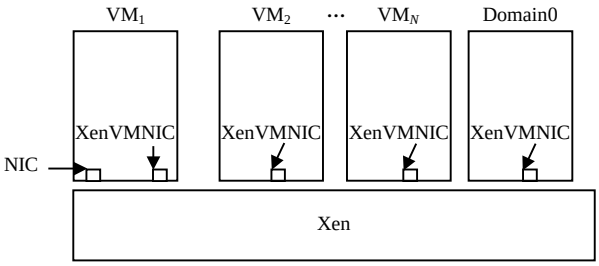

图1　优化模型体系结构

假设 Domain0 总共需要创建和启动 N 个 VM，标记为 $VM_1,VM_2,\cdots,VM_N$，为方便起见，记 VM_0=Domain0。一组服务中的每一个服务单独运行在一个虚拟机中，Xen 为这些服务提供隔离。

优化模型的特点有：

(1) Xen 作为通信介质，对应于 OSI 中 7 层模型的物理层。

(2) 每个 VM 都有一个虚拟网卡，被称为 XenVMNIC，对应于OSI中 7 层模型的链路层。

(3) 各 VM 通过其 XenVMNIC 互联成一个计算机网络，称为 XenVMNet。

(4) 虚拟网卡没有前端和后端的概念，也不需要 Domain0 创建并启动一个专门的DVM。相反，有一个特殊的名为VM_1 的VM。VM_1 至少有两个网卡：一个是Xen的宿主机的物理网卡，与宿主机以外的其他计算机相连；另一个是XenVMNIC，与其他VM相连。除了VM_1，所有其他VM中没有一个能与宿主机以外的计算机相连的网卡。与标准模型相比，优化模型中的虚拟机间直接通信，不需DVM的介入，也不需要网桥设备的转发操作。

(5) 为方便通信双方转移该页的所有权，标准和优化两个模型中的 GOS 都要为每帧分配一页大小的套节字缓冲块。XenVMNIC 的最长帧是 4 096 B，而标准模型的对应值是 1 500 B，因此，新模型减少了要转移的帧数。

(6) 优化模型的 4 096 B 最长帧还显著减少了每个套节字缓冲池内部碎片，显著降低了用于内部碎片合并的时间。

(7) XenVMNIC 有 TCP 效验和卸载功能(TCO)，使得它通知上层 TCP／IP 协议栈不要计算效验和。

进一步，因为通信双方通过共享内存进行数据

传输的误码率很低，设计也取消了XenVMNIC链路层效验和的计算。

XenVMNIC的创建是在VM启动并初始化其GOS装载前进行的。算法伪码如下：

启动Domain0，它创建一个网络接口设备XenVMNIC0

For(对所有 N 个需要创建和启动的 VM)

{

Domain0 创建第 i 个 VM_i 的网络接口设备 $XenVMNIC_i$，i=1,2,…,N

Domain0 创建虚拟机间事件通道 IEC(VM_j, VM_i)，j=0,1,…,i–1；

Domain0 创建虚拟机间共享内存通信环 SMR (VM_j, VM_i)，j=0,1,…,i–1；

}

在 N 个用户 VM 启动以后，Xen 上总共运行 N+1 个 VM。每个 VM 有一个网络接口设备 XenVMNIC，每个 XenVMNIC 有 N 个中断源，这些中断源能被另外的 N 个 VM 所分别触发；每个 XenVMNIC 也有 N 个 SMR，它们分别与另外 N 个 VM 一一对应。

2.1 XenVMNIC帧

XenVMNIC 实现了 XenVMNet 网络的链路层，一个 XenVMNet 帧最长 4 096 B，有如下字段：(1)“VM 标识”字段 2 个字节，标识接收 XenVMNet 帧的目标 VM；(2) “协议类型”字段 2 个字节，标识 XenVMNet 帧所携带的网络层协议类型，如 IP 协议类型等；(3) “长度”字段 2 个字节，标识 XenVMNet 帧所携带的净荷长度；(4) “净荷”字段标识 XenVMNet 帧所携带的净荷，最大净荷长度为 4 090 B。

当GOS的IP协议栈准备把IP包递交给链路层的XenVMNIC时，XenVMNIC必须有能力把封装有IP包的XenVMNIC帧发送给正确的接受者。基于XenVMNIC的ARP被重新设计，VM为XenVMNIC维护一张本地地址映射表，每个表项的“IP地址”字段与“VM标识”字段为一一对应关系，表明后者所表示的VM的XenVMNIC配置了前者所表示的IP地址，每个表项还有一个定时器用于该项的更新。

Xen 维护一张全局地址映射表，每个表项的“VM 标识”字段与“IP 地址”字段也为一一对应关系。VM 既能从全局地址映射表读出表项，当 VM 的 IP 地址改变时，还能更新全局地址映射表。

2.2 XenVMNIC的驱动程序

所有 N+1 个 VM 的 GOS 负责设备 XenVMNIC 的初始化，把虚拟网卡 XenVMNIC 初始化为非以太网的新设备，其 MTU 值(最大传输单元)为 4 KB (1 页)。

遵循 GOS 内核与通用网络设备驱动程序的接口规范，编写 XenVMNIC 的驱动程序。

驱动程序的基本功能是使通信双方接收和发送 XenVMNIC 帧，通信双方的通信协议与标准模型中前端和后端的通信协议一致。

驱动程序还包括一些特殊的功能，比如读或写全局地址映射表的例程等。

2.3 实验和评估

实验的硬件平台是一台 PC 服务器，配置为：一个 3.16 GHZ CPU Xeon、1 MB L1 cache、3 GB RAM、210 GB 硬盘、1 Gb/s 以太网卡。用 Xen 2.0 作为 VMM，虚拟机 Domain0 和 VM_1 运行带 Linux-2.6.9-xen0 内核映象和 Redhat 9.0 发布的 GOS，所有其他虚拟机运行带 Linux-2.6.9-xenU 内核映象和相同发布的 GOS。

设计的实验评估该原型带来的性能改善和开销。虚拟机配置为：100 MB RAM、4 GB 虚拟硬盘、XenVMNIC 虚拟网卡的 Domain0，以及各有 512 MB RAM、10 GB 虚拟硬盘、XenVMNIC 虚拟网卡的 VM_1、VM_2、VM_3(VM_1 有访问物理网卡的特权)。

用两个测评工具测评 XenVMNet 的性能。

(1) 一个工具是 Netperf[8]，它测量在 XenVMNet 网络上 TCP 和 UDP 的最大吞吐量，如表 1 所示。其中，TCP_DEF、UDP_DEF 分别代表标准模型的 TCP、UDP；TCP_OPT、UDP_OPT 分别代表优化模型的 TCP、UDP。在表 1 的测评结果中，优化模型在 TCP 和 UDP 的吞吐量是标准模型的约 2～3 倍。

表 1 两个模型的吞吐量比较

	接收套节字大小 /byte	发送套节字大小 /byte	吞吐量 /10^6 bits·s^{-1}
TCP_DEF	87 380	16 384	833.48
TCP_OPT	87 380	16 384	2 100.37
UDP_DEF	65 536	65 536	350.60
UDP_OPT	65 536	65 536	957.14

(2) 另一个工具是 TPC-W[9]，其 3 层化实现测量原型 XenVMNet 带来用户请求响应时间的改进。TPC-W-NYU[10]是 TPC-W 的一个 J2EE 实现；HTTP 层由 Apache/Tomcat 实现；JBoss 作为应用服务器，

Mysql 作为数据库服务器。

分别在新模型和优化模型下运行 TPC-W-NYU，比较用户请求响应时间的平均值和最大值，如表 2 所示。TPC-W 测评值表明了平均用户请求响应时间改善了 42%，优化模型改善了 TPC-W 各层间的通信性能，降低了用户请求响应时间。

表 2　两个模型的用户请求响应时间比较

用户请求响应时间	标准模型/ms	优化模型/ms	改善率/(%)
平均值	1 022	720	42
最大值	19 244	11 953	61

在表 1 和表 2 的吞吐测评中，优化模型比标准模型具有更好的网络性能。因为：

(1) 与标准模型相比，优化模型中的VM_2与VM_3直接通信，不需DVM的介入，也不需要网桥设备的转发操作。

(2) 为了方便通信双方转移该页的所有权，两个模型中的 GOS 都为每帧分配一页大小的套节字缓冲块。与标准模型 1 500 B 的最长帧相比，优化模型 XenVMNIC 的对应值是 4 096 B，减少了要处理的帧数。文献[11-12]研究了 Xen 标准网络模型的性能和开销问题，研究表明，1 500 B 的标准模型最长帧造成每套节字缓冲池的显著浪费和内部碎片，其结果是进一步导致缓冲池上限被频繁地超出，又显著地浪费了用于内部碎片合并的时间。

(3) 优化模型没有在 TCP/IP 协议栈和 Xen VMNIC进行效验和计算的开销。

3　结 束 语

总之，针对一类流行的 IP 网络应用，本文优化了同一台宿主机上基于 Xen 的虚拟机间的通信网络。实验表明，优化改善了虚拟机间的通信性能，较为显著地减少了用户的请求响应时间。下一步的研究方向是，继续针对这类应用，在内核的其他方面(如 Xen 的虚拟机调度算法、虚拟磁盘管理等)提出优化方案。

参 考 文 献

[1] BARHAM P, DRAGOVIC B, FRASER K, et al. Xen and the art of virtualization[C]//Proceedings of the 19th ACM Symposium on Operating Systems Principles. New York, USA: ACM, 2003: 164-177.

[2] The Xen Project. Home of Xen[EB/OL]. [2008-07-01]. http://www.xen.org/.

[3] HAND S, HARRIS T, KOTSOVINOS E, et al. Controlling the xenoserver open platform[C]//Proceedings of 2003 IEEE Conference of Open Architectures and Network Programming (OPENARCH 2003). Los Alamitos, CA, USA: IEEE, 2003: 3-11.

[4] 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.
HUAI Jin-peng, LI Qin, HU Chun-ming. Research and design on hypervisor based virtual computing environment [J]. Journal of Software, 2007, 18(8): 2016-2026.

[5] 科学技术部. 863计划信息技术领域2007年度专题课题申请指南[EB/OL]. [2007-10-02]. http://www.most.gov.cn/tztg/200704/t20070405_42562.htm.
The Ministry of Science and Technology of the PRC. Guide to application for special topic on information technology under the national high-tech research and development plan of China (863)[EB/OL]. [2007-10-02]. http://www.most.gov.cn/tztg/200704/t20070405_42562.htm.

[6] BHULAIL S, SIVASUB R, AMANIAN S, et al. Lecture notes in computer science: managing traffic performance in converged networks[M]. Berlin Heidelberg, GER: Springer, 2007.

[7] 黄　文, 谢寄石. 基于J2EE的数据库连接服务[J]. 电子科技大学学报, 2002, 31(1): 67-71.
HUANG Wen, XIE Ji-shi. Database connection service based on J2EE[J]. Journal of University of Electric Science and Technology of China, 2002, 31(1): 67-71.

[8] Netperf Organization. The netperf benchmark[EB/OL]. [2008-07-01]. http://www.netperf.org/netperf/NetperfPage.html.

[9] TPC Organization. TPC-W: Benchmarking an ecommerce solution[EB/OL]. [2008-07-01]. http://www.tpc.org/information/other/techarticles.asp.

[10] Parallel and Distributed Systems Research Group (PDSG), New York University. NYU TPC-W: a J2EE implementation of the TPC-W benchmark[EB/OL]. [2008-07-01]. http://cs.nyu.edu/totok/professional/software/tpcw/tpcw.html.

[11] MENON A, COX A, ZWAENEPOEL W. Optimizing network virtualization in xen[C]//Proceedings of the USENIX Annual Technical Conference USENIX '06. Berkeley, CA, USA: USENIX, 2006: 15-28.

[12] MENON A, SANTOS J, TURNER Y, et al. Diagnosing performance overheads in the xen virtual machine environment[C]//Proceedings of the 1st ACM/USENIX International Conference on Virtual Execution Environments (VEE '05). New York, USA: ACM, 2005: 13-23.

编　辑　蒋　晓