虚拟机技术及其应用

吴梦堉　　计算机科学与技术学院
3180105091　　995862798@qq.com

日期：2020 年 4 月 30 日

摘　要

本文介绍了虚拟机技术的发展背景，简单描述了虚拟机监视器（VMM）实现的核心技术：cpu 虚拟化、内存化虚拟化、I/O 虚拟化。阐述了当前虚拟机技术的安全性问题。最后描述了虚拟机的应用场景。

关键词: 虚拟机，虚拟化技术，系统虚拟机、安全性问题、应用场景

1　引言

1.1　背景

虚拟机技术出现于上世纪 60 年代，当时为了提高对珍贵的计算资源的利用率促使虚拟机技术得到了广泛的研究和应用。到了 80 和 90 年代，多任务多用户操作系统的普及以及硬件成本下降使虚拟机技术无法发挥其优势，于是人们冷却了对它的研究热情。到了本世纪初在计算机硬件强大性能的前提下，如何降低系统成本、提高系统资源利用率、降低管理成本，如何提高安全性和可靠性、增强可移植性以及提高软件开发效率等课题使虚拟机技术的重要性越来越明显，使虚拟机技术重新成为计算机技术研究的焦点之一。这里的虚拟机指的是基于指令体系接口抽象 ISA 的系统虚拟机（区别于基于应用程序接口 ABI 的进程虚拟机）它是通过在现有平台（裸机或操作系统）上增加一个虚拟层——VMM 虚拟机监视器 Virtual machine monitors 或 Hypervisors）来实现的。

1.2　基本的操作系统和体系结构知识

计算机有 4 个主要的结构化部件，分别是处理器（processor）、内存（internal memory）、输入输出模块（I/O）和系统总线。特别地，当处理器只有一个时，会被称为中央处理器（central processing unit, CPU）。如图 1 处理器从存储器中得到指令和数据，输入部件将数据写人存储器，输出部件从内存中读出数据，控制器向数据通路、存储器、输入和输出部件发出命令信号。

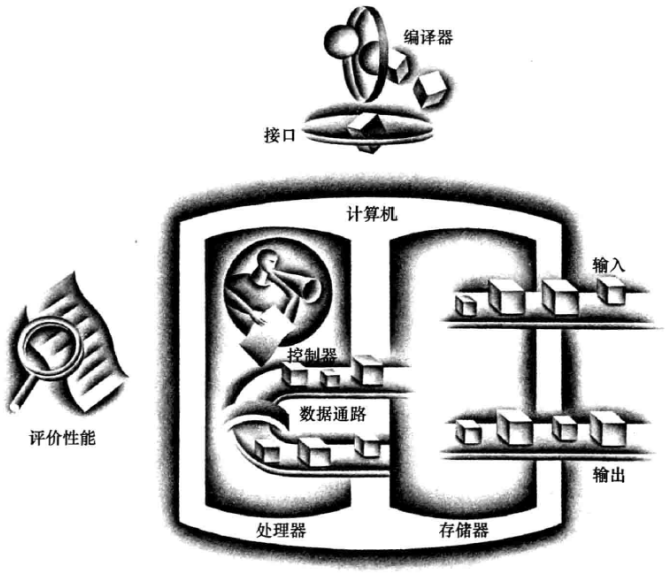

图 1: 组成计算机的 4 个经典部件。[1]

而操作系统（operation system）则作为最底层的软件，有开发并运行程序、访问控制 I/O 设备与文件、访问共享或公共系统等功能。处理器常常会分为用户模式（user mode）和内核模式（kernel mode）（有时也会有监管者模式（supervisor mode））等表示等级不同的模式，在一般情况下用户的进程（process）在用户模式上运行，而操作系统运行内核模式。而操作系统的内存管理机制，在物理内存的基础上建立了空间更多的虚拟内存，处理器中的内存管理单元（memory management unit, MMU）用页表结构维护着虚拟地址（virtual address）到物理地址（physical address）的映射。

图 2: 简化的硬件和软件层次图

2 虚拟化技术概述

2.1 虚拟机

虚拟机（virtual machine）[1]，在计算机科学中的体系结构里，是指一种特殊的软件，可以在计算机平台和终端用户之间创建一种环境，而终端用户则是基于虚拟机这个软件所创建的环境来操作其它软件。虚拟机是计算机系统的仿真器，通过软件模拟具有完整硬件系统功能的、运行在一个完全隔离环境中的完整计算机系统，能提供物理计算机的功能。

按照功能，虚拟机有如下种类：

1. 系统虚拟机（也称为全虚拟化虚拟机）可代替物理计算机。它提供了运行整个操作系统所需的功能。虚拟机监视器（hypervisor）共享和管理硬件，从而允许有相互隔离但存在于同一物理机器上的多个环境。现代虚拟机监视器使用虚拟化专用硬件（主要是主机 CPU）来进行硬件辅助虚拟化。

   例如：VMware VirtualBox
2. 程序虚拟机被设计用来在与平台无关的环境中执行计算机程序。

   例如：Java 虚拟机（JVM）
3. 操作系统层虚拟化

   例如：Docker

2.2 虚拟化技术分类

虚拟化技术大致可分为 5 类：[2]

（1）分区技术（Partition）. 这种虚拟化技术是最早诞生的，原因是当时人们想要提高大型机的利用率。硬件分区技术是将硬件资源划分成数个分区，每个分区享有独立的 CPU、内存，并安装独立的操作系统。

（2）完全虚拟化技术（Full virtualization）. 该技术不再对底层硬件资源进行划分，而是拥有一个统一的宿主系统（Host）. 该宿主可以是一个传统操作系统，也可以是个 MM 其上可以安装多个未经更改的客户操作系统（Guest os）. 其代表实例有 Mare 系列、微软的 Virtual PC/Server 系列等。

（3）泛虚拟化技术（Paa- Virtualization）或准虚拟化技术。这种虚拟技术以 Xen 为代表，它在硬件上覆盖一层 Xen Hypervisor，并需要修改操作系统的内核。

（4）抽象虚拟机的仿真（Emulation of abstract virtual Machine）. 这种虚拟机的典型实例是 Jaa 虚拟机和 P- Code interpreter. 它们在实际的计算机上通过软件模拟来实现，拥有自己的处理器、堆栈、寄存器和相应的指令系统等。

（5）操作系统虚拟化技术 OS with resource contain ers）. 例如 Sun 基于 Solaris 平台的 Container 技术，以及 User Mode linux（M）等。这种虚拟技术的特点是一个单的节点运行着唯一的操作系统实例。通过在这个系统上加装虚拟化平台，可以将系统划分成多个独立且相互隔离的容器，每个容器是一个虚拟的操作系统。

[1]https://zh.wikipedia.org/wiki/

2.3 VMM 实现技术

虚拟机器监视器（virtual machine monitor，VMM）是客户操作系统（guest OS）与硬件之间的“通信层”。而 VMM 可以直接运行在系统硬件上，也可以在宿主操作系统（host OS）上运行，成为操作系统一个进程。

VMM 对底层硬件的所有资源进行抽象，然后把它们重新组合并集成到多个 VM（虚拟机 Virtual Machine）中实现虚拟化。每个 VM 都能够运行独立的操作系统实例。VMM 实现技术主要包括 CPU 的虚拟化、内存的虚拟化、IO 的虚拟化技术。

2.3.1 CPU 虚拟化

现代 CPU 一般具有两种以上的状态，操作系统某些指令只有在高特权级的状态（内核态）下才能被正确执行而一般的应用程序运行在低特权级的状态（用户态）下 VMM 使用一种基本的直接执行 Direct Execution）技术在真实机器上执行 M 指令，而 VMM 控制着 CPU. 这种技术要求将 M 的高特权级代码（一般指操作系统内核）运行在低特权级模式下。以 x86 为例，客户操作系统在 ring 0，1，2 下执行的代码将在 ring1 执行，ring3 下的代码仍然在 ring3 下执行.

2.3.2 内存虚拟化

内存的虚拟化中最传统的技术就是镜像页表。在个完全虚拟化的环境下，VMM 代替客户控制硬件页表而客户只控制客户页表。VMM 将客户在客户页表上请求的更改映射到硬件页表上。页表包含物理地址，对客户来说，页表使用的是伪物理地址（Pseudo- physical Address- es;，对 VMM 来说，镜像页表使用的是机器物理地址。VMM 大多数针对客户页表的镜像工作就是将伪物理地址转译成机器物理地址。当镜像页表与客户页表不同步过期）时，VMM 就进行检测并重新同步。只有客户试图写或访问页发生错误时 VMM 才会检查其镜像页表项是否已过期，此时 VMM 对镜像页表项进行重新同步，然后客户重新尝试访问。

2.3.3 I/O 虚拟化

现代计算机环境中，I/O 设备种类丰富，不同厂商生产的具有不同编程接口的 I/O 设备使虚拟的难度相当 O 大。目前主要有三种针对 I/O 设备的虚拟技术

（1）模拟技术（Emulation）.VMM 必须将设备以一种能被客户软件识别的方式提供给客户操作系统，这样就需要将真实硬件用软件来实现。

2）泛虚拟化技术 Paravirtualization）. 泛虚拟化技术要求对客户软件进行修改，修改后的客户软件使用比普通软硬件接口更高级别的抽象与 MM 直接通信。泛虚拟化技术减少了客户操作系统与 VMM 的通信次数，因此较模拟技术而言提高了性能。

（3）直接分配技术 Direct Assignment）. 有时在一个虚拟环境中需要某个特定的 VM 直接享有一个物理 IO 设备，这就需要应用到直接分配技术。

3 虚拟机的安全性问题

3.1 攻击威胁

虚拟机系统比起传统的计算机系统存在更多的攻击点及安全隐患。通过分析，得到虚拟机的攻击模型如图 2 所示，图中箭头表明所受攻击威胁来源。

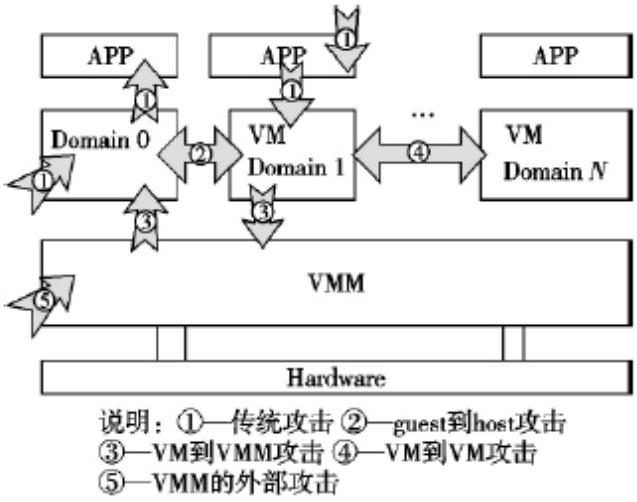

图 3: 虚拟机威胁分析 [3]

3.1.1 VMM 的外部攻击

目前针对 VMM 的外部攻击主要有两种，一种是基于 VM 的 Rootkit 攻击，另一种是恶意代码攻击。

（1）VMBR.[4] 攻击者利用 Rootkit 隐藏自己的踪迹，通过保留 root 访问权限，留下后门的程序集。VMBR 的攻击会在 VMM 的启动之前将程序代码写入内存并运行，一旦攻击者得逞，那么所有虚拟机系统都将在攻击者的控制范围之内。目前比较出名的 VMBR 攻击有 Blue pill 等。检测及防御 VMBR 攻击的方法分析如下：（a）通过计时的方法，有一些指令的执行是通过虚拟出来的，所占用的 CPU 周期会比真实的时间长，可以通过这种方法来进行检测；（b）通过可信模块 TPM 来进行 VMM 的保护的，通过启动过程的完整监测，可以防止 Rootkit 的隐蔽插入，TPM 的设计不但可以抵御 VMBR 的攻击，同时也可以防御其他破坏 VMM 完整性的攻击。

（2）恶意代码。攻击者可以利用远程攻击方法，虚拟机系统的远程管理技术大多是用 HTTP/Https 来连接控制的，因此，MM 必须运行服务器来接受 HTTP 连接。那么攻击者就可以利用 HTTP 的漏洞来进行恶意代码的攻击，如 Xen 的 Xen Api Http 接口就存在 Xss（cross-sIt scripting）漏洞，攻击者可以通过浏览器执行恶意代码脚本。

3.1.2 VM 的 Domain 0 的攻击

Domain0 具有管理其他 VM 的特权，VM 到 Domain0 的攻击体现在 guest-to-host 的攻击来获取 host 的特权，而 VM es- cape 就是这种模式的攻击。VM 通过应用程序，绕过 VMM 的监控而直接访问 Domain 0，从而获取 Domain0 的特权，而旦获取到了 Domain 0 的控制权后，就可以控

制所有 VM. 这些攻击是利用所发现的 bg 来实施的，如 VMware Workstation6 CVE20074496，通过用户授权，进行内存访问和运行恶意代码。另外，VM 还可以利用共享内存通信方式对 VMM 进行病毒分析。

3.1.3 VM 之间的攻击

VMs 之间的攻击体现在通过共同访问的资源来进行恶意攻击。其中隐蔽通道是一个难以解决的问题，攻击者通过进程、内存共享或内存错误，甚至其他错误信息来进行代码的植入和攻击，目前也尚未得到很好的解决。

3.2 其他安全隐患

（1）备份、快照及还原漏洞。在虚拟机中，VMM 提供了备份、快照和还原的功能，一旦系统崩溃了，可以通过快照进行还原，这为系统的维护带来了方便且具有实效性。然而这也导致了新的问题发生：（a）这种机制使得 VMs 容易受到新的攻击，因为许多安全攻击是依赖于线性时间的，重新访问以前的系统 V 状态会违反这些协议；（b）还原后，系统以前存在的漏洞会全部出现，可能没有安全补丁，或旧的安全机制（防火墙规则、反病毒签名等），重新激活先前那些封锁的账号和密码，这都带来了很多的安全隐患。

（2）DMA 攻击。在虚拟中有一种数据传输不受 VMM 控制，这就是 DMA 传输。VM 通过 Domain0 与硬件建立 DMA 连接，而后将数据控制权交由 VM 进行数据传输，在数据传输的过程中，数据将直接从网卡传输到目的 VM 中，在大数据量的传输效率上有很大的提高。然而，这也为攻击者提供了方便，攻击者将轻而易举地利用 DMA 方式将恶意代码或者病毒文件等传入没有安全防范的目标机中，从而达到攻击的目的。

4 虚拟机的应用

4.1 个人使用

用户可以虚拟一台主机以运行不同操作系统和应用程序。桌面虚拟化已经遍布了个人电脑。虚拟机软件使得用户在软件上所花的相关费用可能会越来越低。今后人们还可以方便的创建、复制、携带和共享 VM.

调试器和计算机系统学习。一般来说，只有 ISA 虚拟机才具有系统调试器功能，原因在于它们的可性和可观察性，但它唯一的缺点是运行速度实在太慢

4.2 服务器

随着务器性能的日渐提高，利用虚拟化技术在一台服务器上部署多个操作系统也成为趋势。其优点是可以充分发挥服务器的计算资源而又能使各种应用相隔离。服务器虚拟化减少了硬件采购成本，简化了服务器的部署、管理和维护工作，降低了管理费用，提高了对服务器资源的利用率，还能方便地提供灾难恢复解决方案。

4.3 反病毒

反病毒。目前最常用的两种反病毒技术是特征码技术和虚拟机技术。虚拟机技术的思想是模拟出一台计算机的运行环境，让有可能含有病毒代码的程序在上面运行，通过对运行过程中的异常情况进行监视，若发现些危险行为如自身复制和文件感染等，则很有可能包含病毒代码。出此可以看出，蒜拟机技术是一种启发式探测未知病毒的反病毒技术，能够很有效地测出米知病毒及危险代码因此，虚拟机技术乜是当今反病毒软件的研究热点和发展趋势。

4.4 数据中心化

数据中心管理开始走向虚拟化。随着虚拟化第三阶段的到来，虚拟服务器、虚拟存储、虚拟网络的普及使得数据管理的复杂度将会越来越大，导致需要工具来管理多个技术领域的应用。因此不少厂商纷纷推出数据中心管理工具来管理这些虚拟技术领域以减少管理难度。

参考文献

[1] PATTERSON D A, HENNESSY J L. Computer organization and design arm edition: The hardware software interface [M]. [S.l.]: Morgan kaufmann, 2016.

[2] 黄亭宇, 张琼声, 夏守姬. 系统虚拟机实现技术综述[J]. 农业网络信息, 2007(10):201-204.

[3] 秦中元, 沈日胜, 张群芳, 等. 虚拟机系统安全综述[J]. 计算机应用研究, 2012(29(05)):1618-1622.

[4] KING S T W Y M, CHEN P M. SubVirt: implementing malware with virtual machines[J]. IEEE Computer Society, 2006: 314-327.