高等学校应用型新工科创新人才培养计划指定教材
高等学校云计算与大数据专业“十三五”课改规划教材

“新工科”
配套教材
五大专业集群
教学资源完备

# 云计算与虚拟化技术

青岛英谷教育科技股份有限公司　编著

## 立体化教辅

- 教学PPT
- 教学大纲
- 考试大纲
- 开源代码
- 配套设备
- 在线题库
- 视频讲解

西安电子科技大学出版社
http://www.xduph.com

高等学校应用型新工科创新人才培养计划指定教材

高等学校云计算与大数据专业“十三五”课改规划教材

# 云计算与虚拟化技术

青岛英谷教育科技股份有限公司　编著


西安电子科技大学出版社

## 内 容 简 介

本书以云计算环境下的虚拟化应用为中心，侧重于虚拟化硬件配置、资源限制及系统管理等方面的实用技能培养，旨在为搭建一个完整实用的虚拟化环境提供完备的理论基础和实践指导。

全书共分 11 章。第 1 章简要介绍了虚拟化技术的历史演变、实现原理、常用软件以及与云计算的关系；第 2 章至第 9 章以目前业内最常用的虚拟化软件 KVM 为例，详细讲解了创建 KVM 虚拟机、CPU 虚拟化、内存虚拟化、网络虚拟化、存储虚拟化、资源限制、分布式文件系统管理和管理虚拟机等常用操作；第 10 章和第 11 章对容器虚拟化技术 Docker 进行了专题讲解，包括对 Docker 的基本应用与 Docker 集群高级应用的介绍和指导。

本书内容精练，适用面广，可作为高等院校大数据、计算机科学与技术、软件工程、计算机软件、计算机信息管理等专业的教材，也可作为虚拟化和云计算从业者及爱好者的参考用书。

**图书在版编目(CIP)数据**

**云计算与虚拟化技术** / 青岛英谷教育科技股份有限公司编著
. —西安：西安电子科技大学出版社，2018.2
ISBN 978-7-5606-4834-7

Ⅰ. ① 云… Ⅱ. ① 青… Ⅲ. ① 云计算—研究 ② 数字技术—研究
Ⅳ. ① TP393.027 ② TN01

**中国版本图书馆 CIP 数据核字(2018)第 010577 号**

策　　划　毛红兵
责任编辑　刘玉芳　阎　彬
出版发行　西安电子科技大学出版社(西安市太白南路 2 号)
电　　话　(029)88242885　88201467　　邮　　编　710071
网　　址　www.xduph.com　　电子邮箱　xdupfxb001@163.com
经　　销　新华书店
印刷单位　陕西天意印务有限责任公司
版　　次　2018 年 2 月第 1 版　2018 年 2 月第 1 次印刷
开　　本　787 毫米×1092 毫米　1/16　印　张　22.5
字　　数　531 千字
印　　数　1～3000 册
定　　价　59.00 元
ISBN 978-7-5606-4834-7 / TP
**XDUP 5136001-1**
*** 如有印装问题可调换 ***

# 高等学校云计算与大数据专业“十三五”课改规划教材编委会


主　编　王　燕

副主编　徐凤生　禹继国　赵景秀

编　委　董立华　李吉忠　倪建成　吴海峰

杜永生　孔繁之　王玉锋　张玉坤

董宪武　侯方博　王艳春　葛敬军

王　锋　国　冰　周小双　闫立梅

刘英哲

# ❖❖❖ 前　言 ❖❖❖

随着信息技术的发展，计算机和互联网逐渐融入人们的工作和生活，人们频繁地应用计算机和互联网安排日常出行、就餐购物、休闲娱乐等事宜，数据信息的采集和整理变得越发重要。这时一系列的问题就产生了：如何通过数据信息高效地管理业务？如何动态地调整资源，降低成本？怎样共享已有平台之间的数据信息？显然，传统的计算机架构体系并不足以支撑起大数据时代，而日趋成熟的虚拟化技术和有着广阔前景的云计算技术，将成为实现大数据价值的左膀右臂。

云计算和虚拟化并非捆绑技术，但二者同时使用可以实现优势互补。云计算和虚拟化又是密切相关的，虚拟化是云计算必不可少的支撑技术之一，使用虚拟化技术可以使云计算基础设施的资源部署更灵活。虚拟化也可以引入云计算的理念，为用户提供所需的资源和服务。

虚拟化的概念早在 20 世纪 60 年代就被正式提出，但一直只在大型机上应用。VMware Inc.的首席科学家 Mendel Rosenblum 曾预言："在未来的几年中，虚拟化不再局限于进行简单的资源部署和作为机房使用，它们将提供一个基本构造块，以提高台式机的可移动性、安全性和可用性。"如今，虚拟化技术已臻成熟，在计算机体系结构、操作系统和编程语言等诸多领域都扮演着重要角色：它整合了服务器，将服务器在硬件、功耗、冷却和空间等方面的要求降低了 50%~70%；同时提高了运营效率，最大限度地减少停机的次数，确保了业务的连续性；它还允许快速创建、链接和克隆虚拟机，大大简化了程序开发、测试和上线的过程。

而云计算被视为继 Web 2.0 之后科技产业最重要的商机。2009 年起，阿里、Google、中国移动、HP、微软、IBM、Microsoft、英特尔、戴尔等行业巨头先后推出了自有的云计算服务。2010 年 10 月，《国务院关于加快培育和发展战略性新兴产业的决定》将新一代信息技术产业列入重点发展的七大战略性新兴产业之一，而云计算则是其中的重中之重。同年 10 月 18 日，工信部、发改委联合印发了《关于做好云计算服务创新发展试点示范工作的通知》，在许多城市大力推动云计算产业的发展，近十年来，已经取得了许多显著的成果。

本书是面向高等院校虚拟化与云计算专业的标准化教材，兼顾完善的理论性和较强的实用性。全书以云计算环境下虚拟化技术的应用为核心，第 1 章介绍了虚拟化和云计算的

此为试读，需要完整PDF请访问：www.ertongbook.com

基本原理和常用概念；第 2 章～第 9 章以虚拟化软件 KVM 为基础，重点讲解了 CPU 虚拟化、内存虚拟化、网络虚拟化、存储虚拟化、资源限制、分布式文件系统管理及管理虚拟机的相关知识；最后两章对 Docker 容器虚拟化技术进行专题介绍，主要包括 Docker 基础应用以及 Docker 集群高级应用两部分内容。读者通过对本书的学习，可以了解虚拟化技术的背景和原理，掌握 KVM 下创建虚拟机的方法，掌握 CPU 和内存虚拟化、资源限制和分布式文件系统的应用方法，了解网络虚拟化与存储虚拟化的相关方法，掌握 Docker 的使用方法和技巧，对虚拟化技术拥有一定的理解和实践能力。云计算与虚拟化技术的更新很快，为了能够与最新技术更新保持同步，本书在每章小结处放置了一个“最新更新”二维码，扫描后可及时查看最新知识点，敬请关注。

本书由青岛英谷教育科技股份有限公司编写，参与本书编写工作的有张杰、凌月婷、孟洁、金成学、焦裕朋、张伟洋、侯方超、张玉星、王燕和刘英哲等。本书在编写期间得到了各合作院校专家及一线教师的大力支持，在此，要特别感谢给予我们开发团队大力支持和帮助的领导及同事，感谢合作院校的师生给予我们的支持和鼓励，更要感谢开发团队每一位成员所付出的艰辛劳动。

由于水平所限，书中难免有不当之处，读者在阅读过程中如有发现，可以通过邮箱(yinggu@121ugrow.com)与我们联系，以期不断完善。

本书编委会

2017 年 10 月

# ❖❖❖ 目　录 ❖❖❖

# 第1章　云计算与虚拟化概论

## 本章目标

- 了解虚拟化的演进历史
- 了解虚拟化的实现层次
- 了解常用的虚拟化软件
- 了解容器虚拟化
- 了解 IaaS、PaaS、SaaS 的概念和相互区别
- 了解主流的云平台
- 了解几种流行的开源 IaaS 云平台

虚拟化和云计算技术是当下最为炙手可热的主流 IT 技术。虚拟化技术实现了 IT 资源的逻辑抽象和统一表示，在大规模数据中心管理和解决方案交付等方面发挥着巨大的作用，是支撑云计算伟大构想的最重要的技术基石；而在未来，云计算将会成为计算机的发展趋势和最终目标，以提高资源利用效率，满足用户不断增长的计算需求。云计算以虚拟化为核心，虚拟化则为云计算提供技术支持，二者相互依托，共同发展。

## 1.1 虚拟化简介

虚拟化技术和多任务操作系统的根本目的相同，即让计算机拥有能满足不止一个任务需求的处理能力。近年来，虚拟化技术已成为构建企业 IT 环境的必备技术，许多企业里虚拟机的数量已经远远大于物理机，虚拟化技术已成为 IT 从业者尤其是运维工程师的必备技能。

### 1.1.1 虚拟化技术的起源

1961 年，IBM 公司出品的 IBM709 型计算机(如图 1-1 所示)最早实现了分时系统，该系统将 CPU 的占用切分为多个极短的(1/100sec)时间片，每一个时间片都可以同时执行不同的任务，而通过对这些时间片的交替使用，可以将一个 CPU 虚拟成多个 CPU，并且让每一个虚拟 CPU 看起来都在同时运行，这就是虚拟机的雏形，之后的 System360 型计算机亦支持这一分时系统。

图 1-1 IBM709 型计算机

1972 年，IBM 正式将 System370 型计算机的分时系统命名为虚拟机(Virtual Machine)。

1990 年，IBM 的 System390 型计算机开始支持逻辑分区，允许用户将一个 CPU 分为若干份(最多 10 份)，即可以将一个物理 CPU 分为 10 个逻辑 CPU，且每一个逻辑 CPU 都是独立的。后来 IBM 将这些分时系统开源，才有了 X86 平台上虚拟机的发展。

### 1.1.2 虚拟化技术的原理和特点

虚拟化技术能将许多虚拟服务器融合到一个单独的物理主机上，并通过运行环境的彻底隔离充分保障虚拟机系统的安全。因此，通过虚拟化技术，服务提供方理论上可以为每个客户创建一个独占的虚拟主机。

**1. 虚拟化技术的原理**

虚拟化技术的原理如下：在操作系统中加入一个虚拟化层(Hypervisor)，这是一种位于物理机和操作系统之间的软件，允许多个操作系统共享一套基础硬件，也叫虚拟机监视



此为试读，需要完整PDF请访问：www.ertongbook.com

器(Virtual Machine Monitor，VMM)，该虚拟化层可以对下层主机的物理硬件资源(包括物理 CPU、内存、磁盘、网卡、显卡等)进行封装和隔离，将其抽象为另一种形式的逻辑资源，然后提供给上层虚拟机使用。本质上，虚拟化层是联系主机和虚拟机的一个中间件。

通过虚拟化技术构建的虚拟机一般被称为 GuestOS(客户机)，而作为 GuestOS 载体的物理主机则被称为 HostOS(宿主机)。

一个系统要成为虚拟机，需要满足以下条件：

- ✧ 由 VMM 提供的高效(>80%)、独立的计算机系统。
- ✧ 拥有自己的虚拟硬件(CPU、内存、网络设备、存储设备等)。
- ✧ 上层软件会将该系统识别为真实物理机。
- ✧ 有虚拟机控制台。

**2. 虚拟化技术的特点**

虚拟化技术有以下主要特点：

- ✧ 同质：虚拟机的本质与物理机的本质相同。例如，二者 CPU 的 ISA(指令集架构 Instruction Set Architecture)是相同的。
- ✧ 高效：虚拟机的性能与物理机接近，在虚拟机上执行的大多数指令有直接在硬件上执行的权限和能力，只有少数的敏感指令会由 VMM 来处理。
- ✧ 资源可控：VMM 对物理机和虚拟机的资源都是绝对可控的。
- ✧ 移植方便：如果物理主机发生故障或者因为其他原因需要停机，虚拟机可以迅速移植到其他物理主机上，从而确保生产或者服务不会停止；物理主机故障修复后，还可以迅速移植回去，从而充分利用硬件资源。

Red Hat 公司曾经测试过一些应用服务在虚拟机上运行的效率，部分测试报告如表 1-1 所示。

**表 1-1　虚拟机性能测试报告(节选)**

| IBM DB2 | SAP | ORACLE | JAVA | LAMP |
|---|---|---|---|---|
| VM=HOST*90% | VM=HOST*90% | VM=HOST*90% | VM=HOST*94% | VM=HOST*138% |

从表 1-1 的数据可以看出，该虚拟机的性能已经相当接近物理机，LAMP 服务在虚拟机上运行的效率甚至还提高了，因为虚拟化技术将其中的 Apache、PHP/Python、MySQL 三个应用服务拆分到了三个不同的虚拟机中运行。

## 1.1.3　虚拟化的实现层次

在理解各种虚拟化的实现方法之前，需要先了解 X86 CPU 的一般架构。

为了保证代码执行的安全性、多用户的独立性以及操作系统的正常运行，研发者引入了 CPU 执行状态的概念，用来限制不同程序的访问能力，避免一个程序获取另一个程序的内存数据，并防止程序对物理硬件进行错误操作。

一般而言，CPU 都可以分为用户态和内核态两种基本状态，而 X86 CPU 更是可以细分为 Ring3～0 四种状态，如图 1-2 所示。

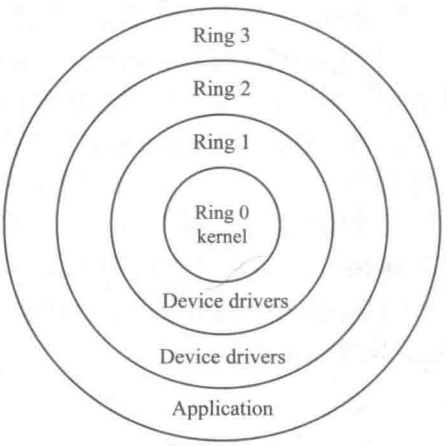


图 1-2　X86 CPU 状态分类

X86 CPU 的各种状态简介如下：

- ✧ Ring0：内核态(Kernel Mode)，宿主机操作系统内核运行的层次，运行在核心态的代码可以无限制地对系统内存、设备驱动程序、网卡接口、显卡接口等外围设备进行访问。
- ✧ Ring1 和 Ring2：驱动层，不涉及应用程序，与虚拟化的实现关系不大。
- ✧ Ring3：用户态(User Mode)，运行在用户态的代码要受到 CPU 的检查，这些代码只能访问内存页表项中允许用户态访问的页面的虚拟地址(受限的内存访问)，而且只能访问 TSS(Task State Segment，指操作系统进程管理过程中，进程切换时的任务现场信息)中的 I/O Permission Bitmap(I/O 许可位图)中规定能被用户态访问的端口，不能访问外围设备，也不能抢占 CPU。所有的用户程序(Application)都运行在用户态，当这些程序需要调用硬件设备时，CPU 会通过专用接口调用核心态的代码，之后这些程序才能对硬件设备进行操作。如果用户态的应用程序直接调用硬件设备，就会被宿主机操作系统捕捉到并触发异常报告。

粗略而言，GuestOS 和 VMM 都属于运行在 Ring3 上的应用程序，GuestOS 操作硬件设备时，会先将操作指令传递给 VMM，由 VMM 对该指令进行监控和检测，然后将指令传递给 HostOS，HostOS 则会将 GuestOS 发出的用户态指令模拟为核心态指令，最后交给 CPU 处理。

### 1.1.4　虚拟化的实现方式

虚拟化的实现方式分为全软件模拟、虚拟化层翻译和容器虚拟化三种。

#### 1. 全软件模拟

全软件模拟技术理论上可以模拟所有已知的硬件，甚至不存在的硬件，但由于是软件模拟方式，所以效率很低，一般仅用于科研，不适合商业化推广。采用这种技术的软件有 Bochs 以及早期的 QEMU 等。

### 2. 虚拟化层翻译

在虚拟化发展的早期，技术主流是借助软件实现的全虚拟化和半虚拟化这两种虚拟化层翻译技术，两种技术各有优缺点，将其灵活应用于不同的环境，可以充分发挥两者的优越性，获得更好的效果。

Intel 领衔的硬件厂商的介入则开启了硬件虚拟化的新时代，从此全虚拟化和半虚拟化技术有了逐渐靠拢的趋势。当下的虚拟化行业已不再以单纯地出售虚拟化软件为主要盈利手段，而是将虚拟化技术整合在了更大、更完善的虚拟化平台解决方案中。

目前，主流的虚拟化层翻译技术有以下几类：

1) 全虚拟化(Fullvirtualization)

使用全虚拟化技术，GuestOS 可直接在 VMM 上运行而不需要对自身做任何修改。全虚拟化的 GuestOS 具有完全的物理机特性，即 VMM 会为 GuestOS 模拟出它需要的所有抽象资源，包括但不限于 CPU、磁盘、内存、网卡、显卡等。全虚拟化技术的实现架构如图 1-3 所示。

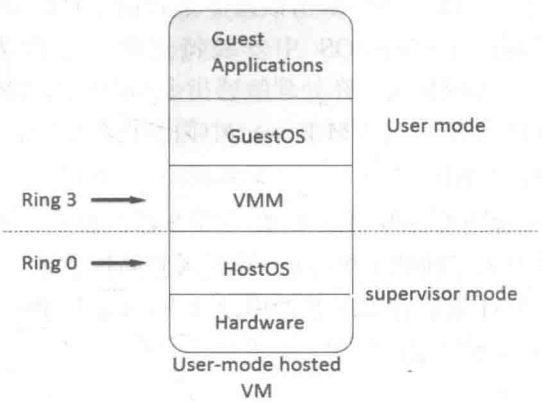


图 1-3　全虚拟化技术架构

用户使用虚拟机 GuestOS 的时候，不可避免会使用 GuestOS 中的虚拟设备驱动程序和核心调度程序来操作硬件设备。例如，GuestOS 使用网卡时，就会调用 VMM 模拟的虚拟网卡驱动来操作物理网卡。但由上图可知，全虚拟化架构下的 GuestOS 是运行在 CPU 用户态中的，因此不能直接操作硬件设备，只有运行在 CPU 核心态中的 HostOS 才可以直接操作硬件设备。为解决这一问题，VMM 引入了特权解除和陷入模拟机制。

(1) 特权解除：又称翻译机制，即当 GuestOS 需要使用运行在核心态的指令时，VMM 就会动态地将该指令捕获，并调用若干运行在非核心态的指令来模拟该核心态指令的效果，从而将核心态的特权解除，解除该特权之后，GuestOS 中的大部分指令都可以正常执行。但是，仅凭借特权解除机制并不能完美解决所有问题，因为在一个 OS 指令集中往往还存在着敏感指令(可能是内核态，也可能是用户态)，这时就需要实现陷入模拟机制。

(2) 陷入模拟：HostOS 和 GuestOS 都存在部分敏感指令(如 reboot、shutdown 等)，这些敏感指令如果被误用会导致很大麻烦。试想，如果在 GuestOS 中执行的“reboot”指令

将 HostOS 重启了，显然会非常糟糕。而 VMM 的陷入模拟机制正是为了解决这个问题：如果在 GuestOS 中执行了需要运行在内核态中的 reboot 指令，则 VMM 首先会将该指令获取、检测并判定为敏感指令，然后启动陷入模拟机制，将敏感指令“reboot”模拟成一个只对 GuestOS 进行操作的、非敏感的，并且运行在非核心态的“reboot”指令，并将其交给 CPU 处理，最后由 CPU 准确执行重启 GuestOS 的操作。

由于全虚拟化是将非内核态指令模拟成内核态指令再交给 CPU 处理，中间要经过两重转换，因此效率比半虚拟化要低，但优点在于不会修改 GuestOS，所以全虚拟化的 VMM 可以安装绝大部分的操作系统。

典型的全虚拟化软件有 VMWare、Hyper-V、KVM-X86(复杂指令集)等。

2) 半虚拟化(Paravirtualization)

半虚拟化技术是需要 GuestOS 协助的虚拟化技术，因为在半虚拟化 VMM 中运行的 GuestOS 内核都经过了修改。一种方式是修改 GuestOS 内核指令集中包括敏感指令在内的内核态指令，使 HostOS 在接收到没有经过半虚拟化 VMM 模拟和翻译处理的 GuestOS 内核态指令或敏感指令时，可以准确判断出该指令是否属于 GuestOS，从而高效地避免错误；另一种方式是在每一个 GuestOS 中安装特定的半虚拟化软件，如 VMTools、RHEVTools 等。因此，半虚拟化技术在处理敏感指令和内核态指令的流程上更为简单。

典型的半虚拟化软件如 Xen、KVM-PowerPC(简易指令集)等。

3) 硬件辅助虚拟化(HVM)

2005 年，Intel 公司提出并开发了由 CPU 直接支持的虚拟化技术，即硬件辅助虚拟化技术，这种虚拟化技术引入了新的 CPU 运行模式和新的指令集，使 VMM 和 GuestOS 运行在不同的模式之下(VMM 运行在 Ring0 的根模式下；GuestOS 则运行在 Ring0 的非根模式下)。硬件辅助虚拟化技术的架构如图 1-4 所示。

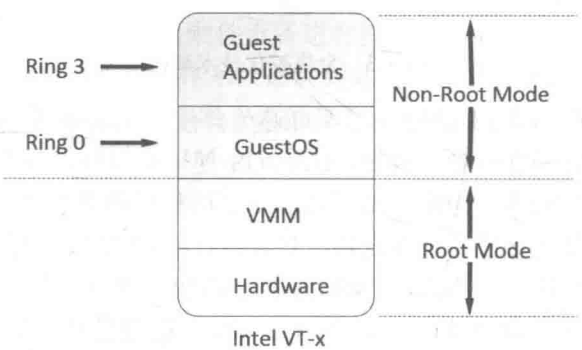


图 1-4　硬件辅助虚拟化技术的架构

CPU 硬件辅助虚拟化技术解决了非内核态敏感指令的陷入模拟难题：由于 GuestOS 运行于受控模式，其内核指令集中的敏感指令会全部陷入 VMM，由 VMM 进行模拟。模式切换时上下文的保存恢复工作也都由硬件来完成，从而大大提高了陷入模拟的效率。硬件辅助虚拟化技术的引入，使 X86 CPU 可以轻松实现完全的虚拟化，因而被之前存在分歧的几乎所有虚拟化软件所采用，包括 KVM-X86、VMWare ESX Server 3、Xen 3.0 等。

目前，主流的硬件辅助虚拟化技术有以下两种：

(1) Intel VT-x：Intel 的 CPU 硬件辅助虚拟化技术，包括 Intel VT Flex-Priority(Intel 灵活任务优先级)、Intel VTF lex-Migration(Intel 虚拟化灵活迁移技术)和 Extended Page-Tables(Intel VT 扩展页表)三大组成部分。Intel VT-x 技术可以让一个 CPU 模拟多个 CPU 的并行运行，从而使一台物理服务器内可以同时运行多个操作系统，降低(甚至消除)多台虚拟机之间的资源争夺和限制，从硬件上极大地改善虚拟机的安全性和性能，提高基于软件的虚拟化解决方案的灵活性与稳定性。

(2) AMD-V：AMD 的 CPU 硬件辅助虚拟化技术，是针对 X86 处理器系统架构开发的一组硬件扩展和硬件辅助虚拟化技术，能够简化基于软件的虚拟化解决方案，改进 VMM 的设计，从而更充分地利用硬件资源，提高服务器和数据中心的虚拟化效率。

4) 内存虚拟化

以往，GuestOS 使用的是虚拟内存，因此必须进行虚拟内存到物理内存的翻译，这就影响了虚拟机的效率，而如今 Intel 的 EPT 技术、AMD 的 RVI 技术均支持内存虚拟化。

内存虚拟化的映射(内存地址转换)涉及以下三类地址：

✧ 虚拟地址(VA)：GuestOS 提供给其应用程序使用的线性地址空间。

✧ 物理地址(PA)：经 VMM 抽象的，虚拟机看到的伪物理地址。

✧ 机器地址(MA)：真实的机器物理地址，即地址总线上出现的地址信号。

宿主机到虚拟机的内存地址映射关系如下：

✧ GuestOS：GuestOS 负责 VA 到 PA 的映射，表达式为 PA = f(VA)。

✧ VMM：VMM 负责 PA 到 MA 的映射，表达式为 MA = g(PA)。

通过上述映射，在应用程序访问地址 VA1 时，该地址会先经过 GuestOS 的页表转换为 PA1，再经过 VMM 的页表将 PA1 转换为 MA1，从而实现了内存地址从虚拟地址到机器地址的转换。

5) 总线虚拟化

总线虚拟化技术可以将一块网卡分给若干个 GuestOS 使用，每台虚拟机分得网卡性能的 1/N。由于总线虚拟化技术是直接把物理设备划分给 GuestOS 的，无需经过 VMM，因此性能较高，甚至接近真机。内存虚拟化和总线虚拟化技术的实现，进一步提高了 GuestOS 和 HostOS 的运行性能。

目前，主流的总线虚拟化技术主要有以下两类：

(1) Intel 的 VT-d 技术。使用 VT-d 技术时，每个 I/O 设备在系统内存中都有一个专用区域，只有该 I/O 设备与分配到该设备的虚拟机才能对该内存区域进行访问。VT-d 技术通过将特定的 I/O 设备安全分配给特定的虚拟机来减少 VMM 管理 I/O 流量的工作，加速了数据传输的速度。

(2) AMD 的 IOMMU 技术。IOMMU 位于外围设备和主机之间，负责管理设备对系统内存的访问，将设备所请求的地址转换为系统内存地址，并检查每个接入的权限。IOMMU 通过允许 VMM 直接将真实设备分配给客户操作系统的方法来让 I/O 虚拟化更有效，此时，VMM 会将客户请求转换为主机操作系统的真实驱动程序请求。通常情况下，IOMMU 会被部署为 HyperTransport 或者 PCI 桥接设备的一部分。

### 3. 容器虚拟化

容器虚拟化是一种不同于虚拟机方式的虚拟化技术，它并不是一种硬件虚拟化方法，而是一个操作系统级的虚拟化方法，因而不属于全虚拟化和半虚拟化中的任意一类。

容器虚拟化技术以容器为虚拟化的载体单位，容器可以为应用程序提供隔离的运行空间，且一个容器内的变动不会影响其他容器。与虚拟机相比，容器具有以下特性：

- ✧ 在容器里运行的一般是不完整的操作系统(虽然也可以)，而在虚拟机上必须运行完整的操作系统。
- ✧ 容器比虚拟机使用更少的资源，包括 CPU、内存等。
- ✧ 容器在云硬件(或虚拟机)中可被复用，就像虚拟机在裸机上可被复用一样。
- ✧ 容器的部署时间可以短到毫秒级，虚拟机则最少是分钟级。

容器比虚拟机更轻量、效率更高，部署也更加便捷，但容器是将应用打包并以进程的形式运行在操作系统上的，因此应用之间并非完全隔离，这是容器虚拟化的一大缺陷。

Linux 系统上的容器虚拟化技术被称为 LXC(Linux Container)，是最常使用的容器虚拟化方案之一，本书第 10 章与第 11 章会以优化的 LXC 技术——Docker 技术为例，对容器虚拟化方法进行专题介绍。

## 1.1.5 常用的虚拟化软件

目前，X86 平台上的主流虚拟化软件可分为两类：面向企业的 VMware、Hyper-V、Xen、KVM，以及面向个人用户的 VMware WorkStation 和 Virtual-Box。

### 1. VMware

VMware 首款产品发布于 1999 年，是最早出现在 X86 平台上的虚拟化软件，具有良好的兼容性和稳定性。VMware 的产品线较为全面，既有虚拟化整体解决方案，也有 IaaS、PaaS、SaaS 平台以及网络、存储等方面的产品。经过近 20 年的发展，VMware 的专用协议得到了很多厂商的支持，已经形成了自己的产品生态链。

VMware 的软件是非开源产品，对用户收费，所以一般只被传统行业与政府机关所采用，中小企业和互联网公司使用较少。

### 2. Xen

Xen 由剑桥大学开发，是最早的开源虚拟化软件，半虚拟化的概念也是 Xen 最早提出的。2007 年 8 月，Xen 被 Citrix 公司收购，并发布了管理工具 XenServer。

作为一种比较古老的虚拟化软件，Xen 已很少被新建虚拟化系统的公司使用，但由于其兼容性和稳定性较好，一些在 Xen 上有技术积累的公司目前仍在使用这一软件。

### 3. Hyper-V

Hyper-V 是微软出品的虚拟化软件，近年来发展较为迅速。Hyper-V 软件必须在 64 位的 Windows 系统上运行，但也可以创建 Linux 虚拟机。

Hyper-V 是非开源的收费产品，管理工具 SCVMM 的配置比较复杂，在管理多台宿主机时，需要先配置 Windows 域和 Windows Server 集群，因此只在一些 Windows 系统为主的企业中应用。



此为试读，需要完整PDF请访问：www.ertongbook.com