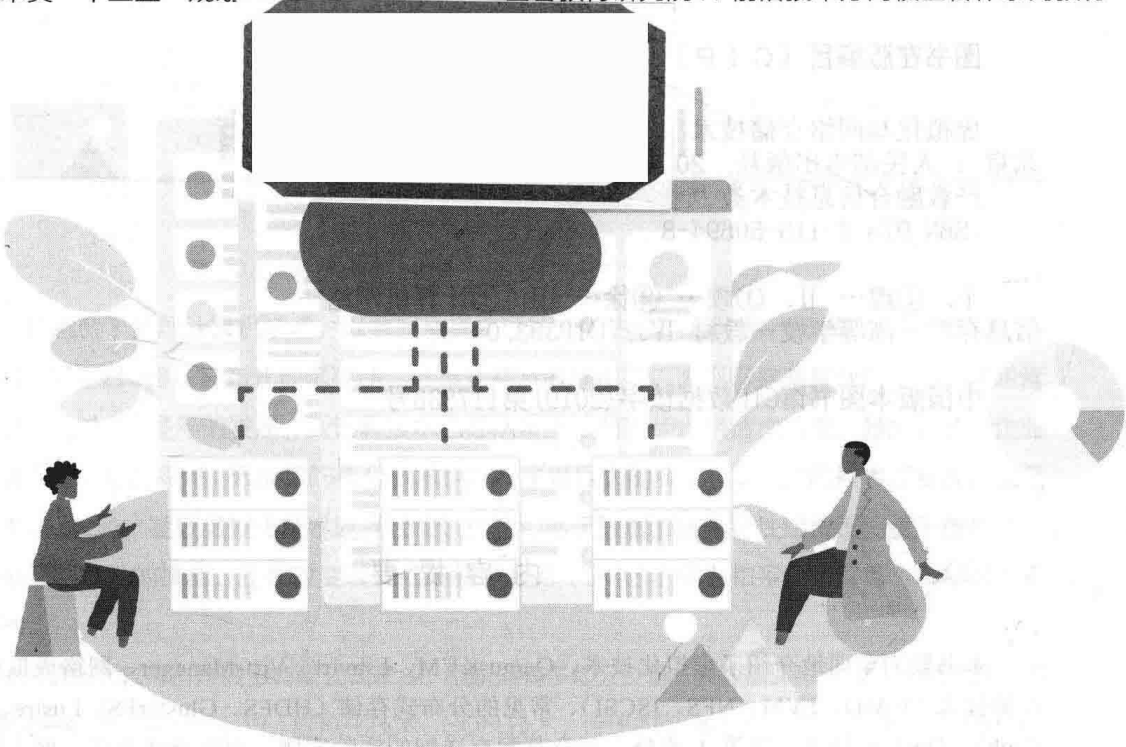


# 虚拟化与 网络存储技术

顾军林 徐义晗 ◎ 主编  
米洪 朱晓彦 林雪纲 ◎ 副主编

## Virtualization and Network Storage Technology

- ◆ 全书采用模块化结构，力求由浅入深，循序渐进
- ◆ 采用“任务驱动”的编写方式，以实际问题引出相关原理和概念
- ◆ 注重教学方法的运用，重视实验教学



# 虚拟化与 网络存储技术

顾军林 徐义晗 ◎ 主编

米洪 朱晓彦 林雪纲 ◎ 副主编

Virtualization and Network  
Storage Technology

人民邮电出版社  
北京

## 图书在版编目(CIP)数据

虚拟化与网络存储技术 / 顾军林, 徐义晗主编. --  
北京: 人民邮电出版社, 2019. 8  
产教融合信息技术类“十三五”规划教材  
ISBN 978-7-115-50694-8

I. ①虚… II. ①顾… ②徐… III. ①计算机网络—  
信息存贮—高等学校—教材 IV. ①TP393.0

中国版本图书馆CIP数据核字(2019)第117835号

## 内 容 提 要

本书较为全面地介绍了虚拟化技术、Qemu-KVM、Libvirt、Virt-Manager、网络虚拟化、传统的存储技术(RAID、LVM、NFS、ISCSI)、常见的分布式存储(HDFS、GlusterFS、Lustre、MooseFS、Ceph)、Docker 技术。除第 1 章外, 每章都配有详细的实验案例, 内容设计丰富, 便于读者理解和掌握。

本书可以作为计算机相关专业的教材, 也可以作为云计算基础入门的培训班教材, 还适合云计算相关从业人员和广大计算机爱好者自学使用。

- 
- ◆ 主 编 顾军林 徐义晗
  - 副 主 编 米 洪 朱晓彦 林雪纲
  - 责任编辑 左仲海
  - 责任印制 马振武
  - ◆ 人民邮电出版社出版发行 北京市丰台区成寿寺路 11 号  
邮编 100164 电子邮件 315@ptpress.com.cn  
网址 <http://www.ptpress.com.cn>  
大厂聚鑫印刷有限责任公司印刷
  - ◆ 开本: 787×1092 1/16  
印张: 17.25 2019 年 8 月第 1 版  
字数: 519 千字 2019 年 8 月河北第 1 次印刷
- 

定价: 56.00 元

读者服务热线: (010)81055256 印装质量热线: (010)81055316

反盗版热线: (010)81055315

广告经营许可证: 京东工商广登字 20170147 号

# 前言 FOREWORD

云计算是一种融合了多项计算机技术，以数据及其处理能力为中心的密集型计算模式，其中以 KVM 虚拟化、SDN 软件定义网络、分布式数据存储、分布式并发编程模型、大规模数据管理和分布式资源管理技术最为关键。经过 10 多年的发展，云计算技术已经从发展培育期步入快速成长期，越来越多的企业开始使用云计算服务。与此同时，云计算的核心技术也在发生着巨大的变化，新一代的技术正在改进甚至取代前一代技术。Docker 容器虚拟化技术以其轻便、灵活和易于快速部署等特性给传统的基于虚拟机的虚拟化技术带来了颠覆性的挑战，正在改变着基础设施即服务（IAAS）平台和平台即服务（PAAS）平台的架构和实现。

本书由北京西普阳光教育科技股份有限公司牵头，组织拥有多年云计算、大数据课程教学经验的教师共同编写而成。本书详细分析了主流的 KVM 虚拟化技术、SDN 技术、传统的存储技术、主流的分布式存储技术、Docker 容器技术等，包含了基础概念分析和详细的实验操作过程，使得学生能全面掌握虚拟化及网络存储相关知识，为后续学习云计算基础架构平台——OpenStack 开源项目打下坚实的基础。

本书遵循以项目为驱动、任务为目标的编写思路。每个实验项目分为 3 个子内容，第 1 部分提出具体的任务要求，明确做什么；第 2 部分分析任务相关的知识及内容；第 3 部分介绍完成任务的具体操作步骤，做到基础知识介绍有针对性，任务目标操作具体化。

本书的参考学时为 64 学时，建议采用理论实践一体化教学模式，各章的参考学时如下。

章节	课程内容（基本实验+拓展实验）	学时
第 1 章	虚拟化技术	2
第 2 章	Qemu-KVM	6
第 3 章	Libvirt	10
第 4 章	Virt-Manager	8
第 5 章	网络虚拟化	10
第 6 章	传统的存储技术	8
第 7 章	常见的分布式存储	8
第 8 章	Docker 技术	10
	课程考评	2
学时总计		64

本书由淮安信息职业技术学院的顾军林、徐义晗担任主编，南京交通职业技术学院米洪、安徽工业经济职业技术学院朱晓彦、北京西普阳光教育科技股份有限公司林雪钢担任副主编，北京西普阳光教育科技股份有限公司的工程师参与了本书实验内容的测试工作，在此表示衷心的感谢。本书配套的 PPT 课件、基本实验和拓

展实验详细步骤视频等资源，读者可以联系北京西普阳光教育科技股份有限公司获得，或登录人民邮电出版社教育社区（[www.ryjiaoyu.com](http://www.ryjiaoyu.com)）下载使用。

由于作者水平有限，书中疏漏和不足之处在所难免，殷切希望广大读者批评指正。同时，恳请读者一旦发现错误，于百忙之中及时与编者联系，以便尽快更正，编者将不胜感激，E-mail: [junlin82@qq.com](mailto:junlin82@qq.com)。

编者

2018年10月

# 目录 CONTENTS

## 第 1 章

<b>虚拟化技术</b> .....	<b>1</b>
1.1 虚拟化技术分类.....	1
1.1.1 CPU 虚拟化.....	2
1.1.2 服务器虚拟化.....	3
1.1.3 存储虚拟化.....	5
1.1.4 网络虚拟化.....	6
1.1.5 应用虚拟化.....	7
1.2 Xen 虚拟化技术简介.....	7
1.2.1 Xen 的历史.....	7
1.2.2 Xen 功能概览.....	8
1.2.3 Xen 虚拟化技术的优点.....	9
1.2.4 Xen 虚拟化技术的缺点.....	9
1.3 KVM 虚拟化技术简介.....	9
1.3.1 KVM 的历史.....	9
1.3.2 KVM 功能概览.....	10
1.3.3 KVM 的优势.....	11
1.3.4 KVM 虚拟化技术的未来.....	11
1.4 Red Hat RHEV 虚拟化系统简介.....	12
1.4.1 RHEV 简介.....	12
1.4.2 RHEV 支持的功能.....	12
1.4.3 RHEV 与 KVM 的区别.....	12
1.4.4 RHEV 的组成.....	13
1.4.5 RHEV 架构.....	14
1.4.6 RHEV 中的资源.....	15
1.4.7 RHEV 虚拟化技术的优点.....	16
1.4.8 RHEV 虚拟化技术的缺点.....	16
1.5 其他虚拟化技术介绍.....	16
1.5.1 VMware.....	16
1.5.2 VirtualBox.....	17
1.5.3 Hyper-V.....	17
1.6 本章小结.....	18

## 第2章

<b>Qemu-KVM .....</b>	<b>19</b>
2.1 KVM 原理简介.....	19
2.1.1 KVM 工作流程.....	19
2.1.2 KVM 架构.....	19
2.1.3 KVM 模块.....	21
2.2 Qemu 原理介绍.....	22
2.2.1 Qemu 架构.....	22
2.2.2 Qemu 模块.....	22
2.2.3 Qemu 的 3 种运行模式.....	23
2.2.4 Qemu 的特点.....	23
2.3 KVM 和 Qemu 的关系.....	24
2.4 Qemu 工具介绍.....	25
2.4.1 Qemu-img.....	25
2.4.2 Qemu-KVM.....	28
2.4.3 Qemu-GA.....	31
2.4.4 Qemu-IO.....	31
2.4.5 Qemu-NBD.....	31
2.5 Qemu 支持的硬盘格式介绍.....	32
【实验 1】 Qemu-KVM 虚拟化环境搭建.....	33
【实验 2】 Qemu-img 生产虚拟机硬盘.....	41
【实验 3】 Qemu-KVM 命令创建虚拟机.....	41
2.6 本章小结.....	45

## 第3章

<b>Libvirt.....</b>	<b>46</b>
3.1 Libvirt 简介.....	46
3.2 Libvirt 简单架构原理介绍.....	47
3.2.1 Libvirt 架构.....	47
3.2.2 Libvirt 运行原理.....	48
3.3 Libvirt API 介绍.....	49
3.3.1 Libvirt API 简介.....	49
3.3.2 与 Hypervisor 建立连接.....	51
3.4 Libvirt 工具集介绍.....	54

3.4.1 Libvirt 安装	54
3.4.2 Libvirt 的配置	56
3.4.3 Libvirtd 的使用	58
3.4.4 Virsh	59
3.5 Libvirt XML 配置文件介绍	62
3.5.1 客户机 XML 配置文件格式示例	62
3.5.2 CPU、内存、启动顺序等基本配置	65
3.5.3 网络的配置	67
3.5.4 存储的配置	69
3.5.5 其他配置简介	70
【实验 4】使用 virsh 创建虚拟机	72
【实验 5】virsh 命令行工具虚拟机的管理	78
【实验 6】virsh 命令行工具网络的管理	81
【实验 7】virsh 命令行工具存储池的管理	88
【实验 8】virsh 命令行工具存储卷的管理	92
3.6 本章小结	95

## 第 4 章

<b>Virt-Manager</b>	<b>96</b>
4.1 Virt-Manager 简介	96
4.2 Virt-Manager 安装	97
4.2.1 环境准备	97
4.2.2 检查 Qemu-KVM、Libvirt 服务	97
4.2.3 检查 VNC 服务的运行	97
4.2.4 安装 Virt-Manager	98
4.3 Virt-Manager 使用介绍	98
4.3.1 打开 Virt-Manager	98
4.3.2 连接至远程 Virt-Manager	99
4.4 WebVirtMgr 介绍	101
4.4.1 WebVirtMgr 管理平台介绍	101
4.4.2 WebVirtMgr 的主要功能	101
【实验 9】使用 Virt-Install 安装虚拟机并使用 Virt-Viewer 连接桌面	102
【实验 10】使用 Virt-Manager 创建虚拟机（在 KVM 上安装 CentOS 7 虚拟机）	105
【实验 11】使用 Virt-Manager 管理存储和网络	109
【实验 12】WebVirtMgr 安装	123
【实验 13】WebVirtMgr 使用	123

4.5 本章小节 ..... 124

## 第 5 章

### 网络虚拟化 ..... 125

5.1 网络虚拟化的驱动力与关键需求 ..... 125

- 5.1.1 网络虚拟化的驱动力 ..... 125
- 5.1.2 网络虚拟化的关键需求 ..... 126
- 5.1.3 软件定义网络 SDN ..... 127

5.2 软件 Overlay SDN 网络, L2/L3 网络 ..... 128

- 5.2.1 Open vSwitch ..... 128
- 5.2.2 Overlay L2/L3 数据流 ..... 129

5.3 硬件 Underlay SDN 网络 ..... 130

5.4 软件化 L4~L7 网络功能 ..... 131

- 5.4.1 L4~L7 网络功能 ..... 131
- 5.4.2 OpenStack Neutron 的 L4~L7 控制面 ..... 132

5.5 网络虚拟化端到端解决方案 ..... 132

- 5.5.1 端到端关键需求 ..... 132
- 5.5.2 端到端解决方案 ..... 133

【实验 14】 Open vSwitch 安装部署 ..... 133

【实验 15】 Net Namespace 综合实验 ..... 138

【实验 16】 OVS 创建 VLAN 虚拟二层环境 ..... 141

【实验 17】 OVS 创建 GRE 隧道网络 ..... 146

【实验 18】 Brctl 搭建 Linux 网桥 ..... 150

5.6 本章小结 ..... 151

## 第 6 章

### 传统的存储技术 ..... 152

6.1 传统存储技术的分类 ..... 152

- 6.1.1 概述 ..... 152
- 6.1.2 存储区域网络 ..... 152

6.2 硬盘结构及接口介绍 ..... 155

- 6.2.1 硬盘结构 ..... 155
- 6.2.2 硬盘的读写 ..... 156
- 6.2.3 硬盘接口 ..... 157

6.3 RAID 技术介绍 ..... 160

6.3.1 RAID 基础知识	160
6.3.2 RAID 的实现方案	161
6.3.3 RAID 技术术语	161
6.4 RAID 技术的特点	163
【实验 19】 mdadm 工具创建软件 RAID	170
6.5 硬盘与分区	178
6.5.1 硬盘分区概述	178
6.5.2 Linux 的分区规定	178
6.5.3 Linux 文件系统类型简介	180
【实验 20】 硬盘的分区及格式化	181
6.6 逻辑卷技术介绍	186
【实验 21】 逻辑卷组及逻辑卷的管理	187
【实验 22】 搭建 NFS 服务器	193
【实验 23】 搭建 iSCSI 环境	194
6.7 本章小结	195

## 第 7 章

<b>常见的分布式存储</b>	<b>196</b>
7.1 分布式系统介绍	196
7.2 HDFS 分布式存储	197
7.2.1 HDFS 架构	197
7.2.2 HDFS 如何读数据	199
7.2.3 HDFS 如何写数据	200
【实验 24】 HDFS 搭建和使用	201
7.3 GlusterFS 分布式存储	209
7.3.1 GlusterFS 系统概述	209
7.3.2 GlusterFS 架构	210
【实验 25】 GlusterFS 搭建和使用	212
7.4 Lustre 分布式存储	217
7.4.1 Lustre 架构	217
7.4.2 Lustre I/O 特点	218
7.4.3 Lustre 读写数据	219
【实验 26】 Lustre 搭建和使用	220
7.5 MooseFS 分布式存储	225
7.5.1 MooseFS 架构	225
7.5.2 MooseFS 读写数据	226

【实验 27】 MooseFS 搭建和使用 .....	227
7.6 Ceph 分布式存储 .....	228
7.6.1 Ceph 架构 .....	228
7.6.2 Ceph 读写数据 .....	229
7.6.3 Ceph 客户端 .....	230
【实验 28】 Ceph 搭建和使用 .....	230
7.7 本章小结 .....	231

## 第 8 章

<b>Docker 技术 .....</b>	<b>232</b>
8.1 Docker 的基本原理 .....	232
8.1.1 Docker 的起源 .....	232
8.1.2 Docker 引擎 .....	232
8.1.3 Docker 的核心概念 .....	233
【实验 29】 Docker 安装部署 .....	235
【实验 30】 Docker 命令行操作 .....	239
8.2 Dockerfile .....	247
8.2.1 Dockerfile 简介 .....	247
8.2.2 Dockerfile 指令详解 .....	247
【实验 31】 Dockerfile 创建 PHP 镜像 .....	250
8.3 Docker Registry .....	257
8.3.1 Docker 仓库简介 .....	257
8.3.2 私有仓库 .....	257
【实验 32】 Docker Registry 的搭建和使用 .....	258
8.4 Kubernetes 容器云 .....	263
8.4.1 Kubernetes 简介 .....	263
8.4.2 Kubernetes 的核心概念 .....	264
8.4.3 Kubernetes 架构 .....	264
【实验 33】 Kubernetes 搭建和使用 .....	265
8.5 本章小结 .....	266

# 第1章

## 虚拟化技术

# 01

### 学习目标

- 1 熟悉常见的虚拟化技术。
- 2 了解 Xen 虚拟化技术及其优缺点。
- 3 了解 KVM 虚拟化技术。
- 4 了解 Red Hat RHEV 虚拟化系统。

基于内核的虚拟机是开源的系统化模块，从 Linux 2.6.20 内核之后，它集成在 Linux 的各个主要发行版本中。它使用 Linux 自身的调度器进行管理，相对于 Xen 核心代码量少。其中 KVM 是目前云计算 OpenStack 架构底层实现的主要技术来源。本章主要介绍虚拟化技术分类、Xen 虚拟化技术、KVM 虚拟化技术、Red Hat RHEV 虚拟化系统及其他公司典型的虚拟化产品。

## 1.1 虚拟化技术分类

虚拟化是一个广义的术语，是指计算元件在虚拟的基础上而不是真实的基础上运行，是一个旨在简化管理、优化资源的解决方案。

如图 1-1 所示，我们可以将一般的计算模型抽象成一定的物理资源和运行于其之上的计算元件，它们之间通过定义的物理资源接口进行交互。随着计算机硬件技术的发展，物理资源的容量越来越大，价格越来越低。在现有的计算资源基础之上，物理资源已经产生了很大的闲置与浪费。为了充分利用新的物理资源，提高效率，一个比较直接的办法就是更新计算元件以利用更加丰富的物理资源。但是，人们往往出于对稳定性和兼容性的追求，并不情愿频繁地对已经存在的计算元件做大幅度变更。虚拟化技术则另辟蹊径，通过引入一个新的虚拟化层，对下管理真实的物理资源，对上提供虚拟的系统资源，从而实现了在扩大硬件容量的同时，简化软件的重新配置过程。

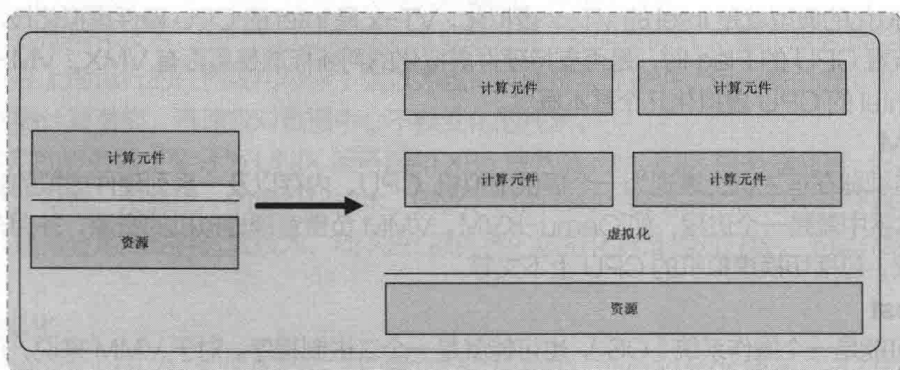


图 1-1 虚拟化逻辑元

为了表述虚拟化的一般概念，图 1-1 使用了资源一词。在实际的应用中，资源可以表现为各种各样的形式。如果把操作系统及其提供的系统调用作为资源，那么虚拟化就表现为操作系统虚拟化。Linux 容器虚拟化技术就是在同样的一个 Linux 操作系统之上，虚拟出多个同样的操作系统。如果将整个 X86 平台包括处理器、内存和外设作为资源，那么对应的虚拟化技术就是平台虚拟化，在同一个 X86 平台上面，可以虚拟出多个 X86 平台，每个虚拟平台都可以运行自己独立完整的操作系统。

虚拟化从本质上讲是指从逻辑角度而不是物理角度来对资源进行配置，它是一种从单一的逻辑角度来看待不同的物理资源的方法。作为一种从逻辑角度出发的资源配置技术，虚拟化是物理实际的逻辑抽象。

对于用户，虚拟化技术实现了软件跟硬件分离，用户不需要考虑后台的具体硬件实现，只需要在虚拟层环境上运行自己的系统和软件。而这些系统和软件在运行时，也似乎跟后台的物理平台无关。下面我们简单介绍一下几种常见的虚拟化技术。

### 1.1.1 CPU 虚拟化

虚拟化在计算机方面通常是指计算元件在虚拟的基础上而不是真实的基础上运行。虚拟化技术可以扩大硬件的容量，简化软件的重新配置过程。简单来说，CPU 的虚拟化技术就是单 CPU 模拟多 CPU 并行，允许一个平台同时运行多个操作系统，并且应用程序可以在相互独立的空间内运行而互不影响，从而显著提高计算机的工作效率。

纯软件虚拟化解方案存在很多限制。“客户”操作系统很多情况下通过虚拟机监视器（Virtual Machine Monitor, VMM）来与硬件进行通信，由 VMM 来决定其对系统上所有虚拟机的访问。（注意，大多数处理器和内存访问独立于 VMM，只在发生特定事件时才会涉及 VMM，如页面错误。）在纯软件虚拟化解方案中，VMM 在软件套件中的位置是传统意义上操作系统所处的位置（如处理器、内存、存储、显卡和网卡等的接口）模拟硬件环境。这种转换必然会增加系统的复杂性。

CPU 的虚拟化技术是一种硬件方案，支持虚拟技术的 CPU 带有特别优化过的指令集来控制虚拟过程，通过这些指令集，VMM 相比软件的虚拟实现方式能很大程度上提高性能。虚拟化技术可提供基于芯片的功能，借助兼容 VMM 软件能够改进纯软件解决方案。虚拟化硬件可提供全新的架构，支持操作系统直接在上面运行，从而无须进行二进制转换，减少了相关的性能开销，极大简化了 VMM 设计，进而使 VMM 能够按通用标准进行编写，性能更加强大。另外，目前在纯软件 VMM 中缺少对 64 位客户操作系统的支持，而随着 64 位处理器的不断普及，这一严重缺点也日益突出。而 CPU 的虚拟化技术除支持广泛的传统操作系统之外，还支持 64 位客户操作系统。

虚拟化技术是一套解决方案。完整的情况需要 CPU、主板芯片组、BIOS 和软件的支持，如 VMM 软件或者某些操作系统本身。即使只有 CPU 支持虚拟化技术，在配合 VMM 的软件情况下，也会比完全不支持虚拟化技术的系统有更好的性能。

CPU 虚拟化的典型就是 Intel 的 VT-x 虚拟化，VT-x 是 Intel 的 CPU 硬件虚拟化技术，但是在操作系统内部查看 CPU 的 Flag 时，是否支持硬件虚拟化的判断标准是是否有 VMX，VMX 是什么，下面简单介绍 Intel 的 CPU 虚拟化几个基本概念。

#### 1. VMM

虚拟机监视器在宿主机上表现为一个提供虚拟机 CPU、内存以及一系列硬件虚拟的实体。这个实体在 KVM 体系中就是一个进程，如 Qemu-KVM。VMM 负责管理虚拟机的资源，并拥有所有虚拟机资源的控制权，包括切换虚拟机的 CPU 上下文等。

#### 2. Guest

Guest 可能是一个操作系统（OS），也可能就是一个二进制程序。对于 VMM 来说，它就是一堆指令集，只需要知道入口（RIP 寄存器值）就可以加载。

Guest 运行需要虚拟 CPU，当 Guest 代码运行的时候，处于 VMX Non-Root 模式。此模式下，该用什么指令还是用什么指令，该用什么寄存器还用用什么寄存器，该用 Cache 还是用 Cache，但是在执行到特殊指令的时候，CPU 控制权即被交给 VMM，由 VMM 来处理特殊指令，完成硬件操作。

### 3. CPU 运行级别

CPU 支持 Ring0~Ring3 四个等级，但是 Linux 只使用了其中的两个——Ring0 和 Ring3。当 CPU 寄存器标示当前 CPU 处于 Ring0 级别的时候，表示此时 CPU 正在运行的是内核的代码。而当 CPU 处于 Ring3 级别的时候，表示此时 CPU 正在运行的是用户级别的代码。当发生系统调用或者进程切换的时候，CPU 会从 Ring3 级别转到 Ring0 级别。Ring3 级别是不允许执行硬件操作的，所有硬件操作都需要内核提供的系统调用来完成。

### 4. VMX

为了从 CPU 层面支持 VT 技术，Intel 在 Ring0~Ring3 的基础上，扩展了传统的 X86 处理器架构，引入了 VMX 模式，VMX 分为 Root 和 Non-Root。VMM 运行在 VMX Root 模式；Guest 运行在 VMX Non-Root 模式。

Intel VT-x 的架构图如图 1-2 所示。

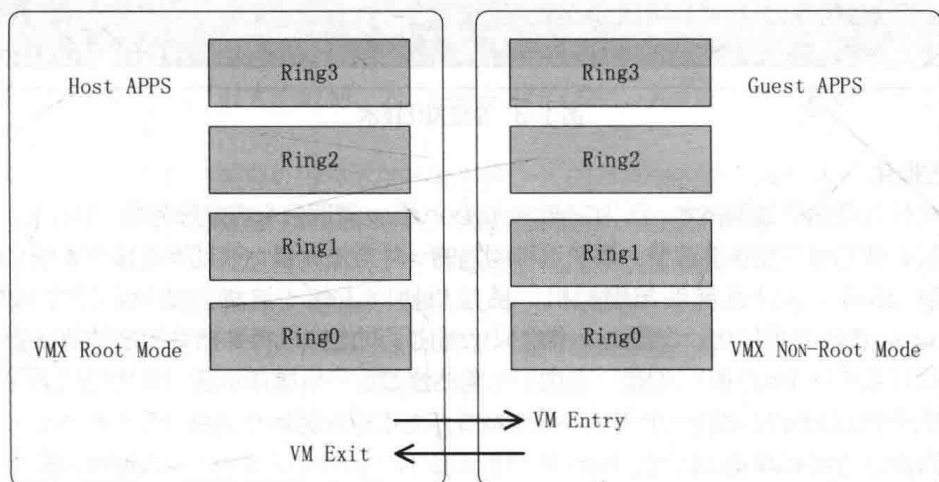


图 1-2 Intel VT-x 架构图

## 1.1.2 服务器虚拟化

服务器虚拟化能够通过区分资源的优先次序，并随时随地将服务器资源分配给最需要它们的工作负载来简化管理和提高效率，从而减少为单个工作负载峰值而储备的资源。

通过服务器虚拟化技术，用户可以动态启用虚拟服务器（又叫虚拟机），每个服务器实际上可以让操作系统（以及在上面运行的任何应用程序）误以为虚拟机就是实际硬件。运行多个虚拟机还可以充分发挥物理服务器的计算潜能，迅速应对数据中心不断变化的需求。

目前，常用的服务器主要分为 UNIX 服务器和 X86 服务器。对 UNIX 服务器而言，IBM、HP、Sun 各有自己的技术标准，没有统一的虚拟化技术。因此，目前 UNIX 服务器的虚拟化还受具体产品平台的制约，不过其通常会用到硬件分区技术。而 X86 服务器的虚拟化标准相对开放。下面介绍 X86 服务器的虚拟化技术。

### 1. 全虚拟化

使用 Hypervisor 在 VM 和底层硬件之间建立一个抽象层，Hypervisor 捕获 CPU 指令，为指令访问硬件控制器和外设充当中介，也为虚拟机的配置提供了最大程度的灵活性。这种虚拟化技术几乎能让

任何一款操作系统不加改动就安装在 VM 上，而操作系统却不知道自己在虚拟化环境下，这也是全虚拟化（Full Virtualization）无可比拟的优势。全虚拟化的主要缺点是 Hypervisor 会带来处理开销。其架构如图 1-3 所示。

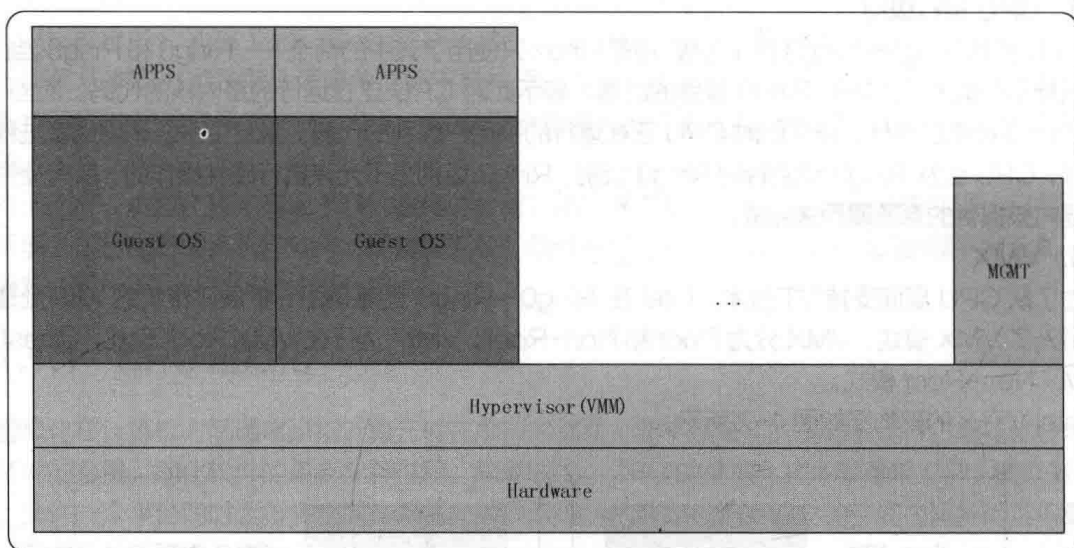


图 1-3 全虚拟化技术

## 2. 准虚拟化

全虚拟化是处理器密集型技术，因为它要求 Hypervisor 管理各个虚拟服务器，并让它们彼此独立，这样会带来没必要的服务器性能浪费。解决这种问题的一种方法就是，改动客户操作系统，使它以为自己运行在虚拟环境下，从而能够与虚拟机、监控机协同工作。这种方法就叫准虚拟化（Para-Virtualization），也叫半虚拟化。本质上，准虚拟化弱化了对虚拟机特殊指令的被动截获要求，将其转化成客户机操作系统的主动通知。但是，准虚拟化需要修改客户机操作系统的源代码来实现主动通知。

Xen 是开源准虚拟化技术的一个例子。操作系统作为虚拟服务器在 Xen Hypervisor 上运行之前，必须在内核层面进行某些改变。因此，Xen 适用于 BSD、Linux、Solaris 及其他开源操作系统，但是不适用像 Windows 这些专有操作系统的虚拟化处理，因为它们不是公开源代码，所以无法修改其内核。其架构如图 1-4 所示。

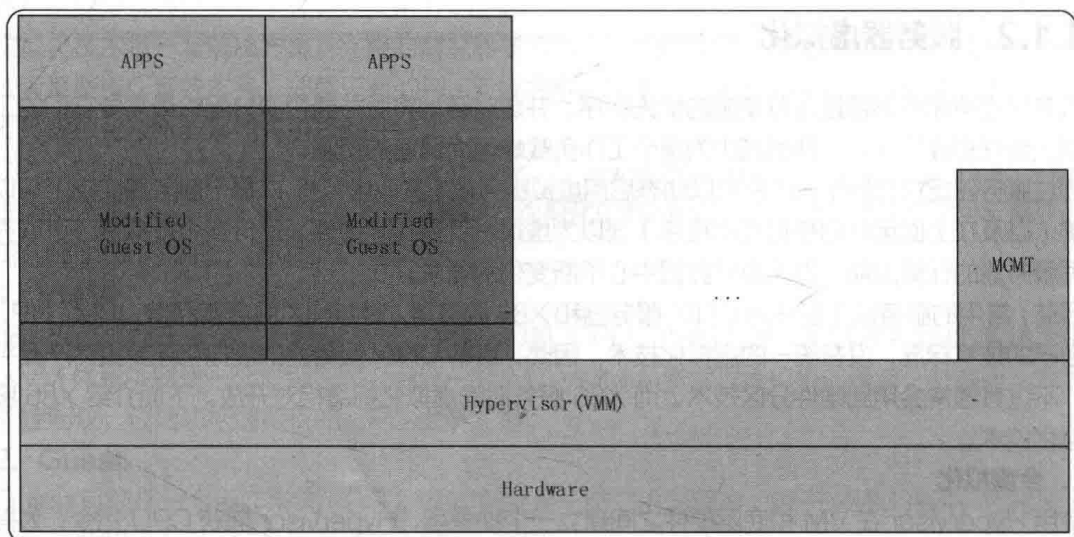


图 1-4 准/半虚拟化技术架构

### 3. 操作系统层虚拟化

实现虚拟化还有一个方法,就是在操作系统层面增添虚拟服务器功能。就操作系统层的虚拟化而言,没有独立的 Hypervisor 层。相反,主机操作系统本身就负责在多个虚拟服务器之间分配硬件资源,并且让这些服务器彼此独立。一个明显的区别是,如果使用操作系统层虚拟化,所有虚拟服务器必须运行同一操作系统。

虽然操作系统层虚拟化的灵活性比较差,但本机速度性能比较高。此外,由于架构在所有虚拟服务器上使用单一、标准的操作系统,因此管理起来比异构环境要容易。操作系统层虚拟化的典型应用便是 Docker,关于 Docker 技术在本书第 8 章会有详细的介绍。

## 1.1.3 存储虚拟化

随着信息业务的不断运行和发展,存储系统网络平台已经成为一个核心平台,大量高价值数据积淀下来。基于对这些数据的应用,人们对平台的要求也越来越高,不光是对于存储容量,还包括对于数据访问性能、数据传输性能、数据管理能力、存储扩展能力等多个方面的要求。可以说,存储网络平台综合性能的优劣,将直接影响整个系统的正常运行。基于这一原因,虚拟化技术又一子领域——虚拟存储技术,应运而生。

其实虚拟化技术并不是一项很新的技术。它的发展应该说是随着计算机技术的发展而发展起来的,其最早始于 20 世纪 70 年代。由于当时的存储容量,特别是内存容量成本非常高,容量也很小,大型应用程序或多程序应用就受到了很大的限制。为了突破这样的限制,人们采用了虚拟存储的技术,最典型的应用就是虚拟内存技术。

随着计算机技术以及相关信息处理技术的不断发展,人们对存储的需求越来越大。这样的需求刺激了各种新技术的出现,结果是硬盘性能越来越高、容量越来越大。但是在大量的大中型信息处理系统中,单个硬盘是不能满足需要的,在这样的情况下,存储虚拟化技术就发展起来了。在这个发展过程中也有几个阶段,出现了几种应用。首先是硬盘条带集(RAID,可带容错)技术,其将多个物理硬盘通过一定的逻辑关系集合起来,成为一个大容量的虚拟硬盘。而随着数据量不断增加和对数据可用性要求的不断提高,又一种新的存储技术应运而生,那就是存储区域网络(SAN)技术。

SAN 的广域化旨在使存储设备实现成为一种公用设施,任何人员、任何主机都可以随时随地获取各自想要的数。目前讨论比较多的包括 iSCSI、FC OVER IP 等技术,虽然一些相关的标准还没有最终确定,但是存储设备公用化、存储网络广域化是一个不可逆转的潮流。存储虚拟化架构如图 1-5 所示。

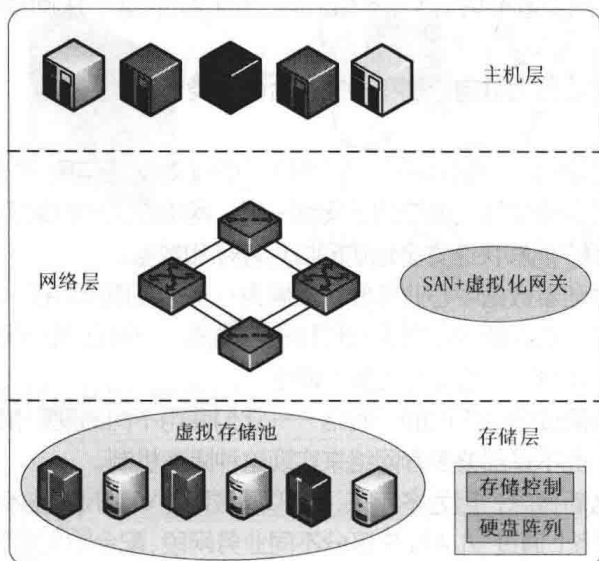


图 1-5 存储虚拟化架构

所谓虚拟存储，就是把多个存储介质模块（如硬盘、RAID）通过一定的手段集中管理起来，所有的存储模块在一个存储池（Storage Pool）中得到统一管理，从主机和工作站的角度，看到的不是多个硬盘，而是一个分区或者卷，就好像是一个超大容量（如1Tb以上）的硬盘。这是一种可以将多种、多个存储设备统一管理起来，为使用者提供大容量、高数据传输性能的存储系统。

虚拟存储设备需要通过大规模的 RAID 子系统和多个 I/O 通道连接到服务器上，智能控制器提供 LUN 访问控制、缓存和其他管理功能如数据复制等。这种方式的优点在于存储设备管理员对设备有完全的控制权，而且通过与服务器系统分开，可以将对存储的管理与多种服务器操作系统隔离，并且可以很容易地调整硬件参数。

从虚拟化存储来讲，拓扑结构主要有两种方式：即对称式（带内管理）与非对称式（带外管理）。对称式虚拟存储技术是指虚拟存储控制设备与存储软件系统、交换设备集成为一个整体，内嵌在网络数据传输路径中；非对称式虚拟存储技术是指虚拟存储控制设备独立于数据传输路径之外。

### 1.1.4 网络虚拟化

网络虚拟化是目前业界关于虚拟化细分领域界定最不明确、存在争议较多的一个概念。微软眼中的“网络虚拟化”，是指虚拟专用网络（VPN）。VPN 对网络连接的概念进行了抽象，允许远程用户访问组织的内部网络，就像物理上连接到该网络一样。网络虚拟化有助于保护 IT 环境，防止来自 Internet 的威胁，同时使用户能够快速安全地访问应用程序和数据。

但是网络巨头思科（Cisco）不那么认为。出身、成名且目前称霸于网络的思科公司，在对 IT 未来的考虑上以网络为核心。它认为在理论上，网络虚拟化能将任何基于服务的传统客户端/服务器安置到“网络上”。这意味着可以让路由器和交换机执行更多的服务，使思科在业界的重要性和营业额都大幅增加。思科认为网络虚拟化由三个部分组成：访问控制、路径提取以及服务优势。从思科的产品规划图上看，该公司的路由器和交换机将拥有诸如安全、存储、VOIP、移动和应用等功能。对思科而言，它们的战略是通过扩大网络基础设备的销售来持续产生盈利。而对用户来讲，这能帮助他们提高网络设备的价值，并调整原有的网络基础设备。

作为网络阵营的另一巨头，3COM 公司在网络虚拟化方面的动作比思科更大。3COM 的路由器中可以插入一张工作卡。该卡上带有一套全功能的 Linux 服务器，可以和路由器中枢相连。在这个 Linux 服务器中，用户可以安装诸如 Sniffer、VOIP 等软件及安全应用等。此外，该公司还计划未来在 Linux 卡上运行 VMware，以支持用户运行 Windows Server。3COM 的这个开源网络虚拟化活动名为 3COM ON（又名开放式网络）。

网络虚拟化从总体来说，分为纵向分割和横向整合两大类。

#### 1. 纵向分割

早期的“网络虚拟化”是指虚拟专用网络（VPN）。VPN 对网络连接的概念进行了抽象，允许远程用户访问组织的内部网络，就像物理上连接到该网络一样。网络虚拟化有助于保护 IT 环境，防止来自 Internet 的威胁，同时使用户能够快速安全地访问应用程序和数据。

随后的网络虚拟化技术随着数据中心业务要求发展为：多种应用承载在一张物理网络上，通过网络虚拟化分割（称为纵向分割）功能使得不同企业机构相互隔离，但可在同一网络上访问自身应用，从而实现将物理网络进行逻辑纵向分割，虚拟化为多个网络。

如果把一个企业网络分隔成多个不同的子网络——它们使用不同的规则和控制，用户就可以充分利用基础网络的虚拟化功能，而不是部署多套网络来实现这种隔离机制。

网络虚拟化并不是什么新概念，因为多年来，虚拟局域网（VLAN）技术作为基本隔离技术已经得到广泛应用。当前在交换网络上通过 VLAN 来区分不同业务网段、配合防火墙等安全产品划分安全区域，是数据中心基本设计内容之一。