

虚拟化与容器技术

陈 涛 主 编

孔繁玉 孙鹏军 张炉焱 副主编

清华大学出版社

北 京

内 容 简 介

本书通过深入浅出的方式介绍 KVM 虚拟化技术与 Docker 容器技术的概念、原理及实现方法，内容包括 KVM 概述、安装 KVM、创建 KVM 虚拟机、虚拟机管理、管理 KVM 虚拟网络、管理 KVM 虚拟存储、容器技术简介、Docker 镜像管理、Docker 容器管理、Docker 网络管理、Docker 存储管理、使用 Dockerfile 创建镜像等，并包含丰富的实验和案例，内容丰富，结构清晰，案例典型，实践性强。

本书既可作为希望了解虚拟化技术和容器技术的系统管理员、DevOps 工程师的技术入门书籍，也可作为高等院校云计算相关课程的教材或教学参考书。

本书封面贴有清华大学出版社防伪标签，无标签者不得销售。

版权所有，侵权必究。举报：010-62782989，beiqinquan@tup.tsinghua.edu.cn。

图书在版编目(CIP)数据

虚拟化与容器技术 / 陈涛主编. —北京：清华大学出版社，2023.9

ISBN 978-7-302-64400-2

I. ①虚… II. ①陈… III. ①Linux 操作系统—虚拟处理机 IV. ①TP338

中国国家版本馆 CIP 数据核字(2023)第 149539 号

责任编辑：刘金喜

封面设计：高娟妮

版式设计：孔祥峰

责任校对：成凤进

责任印制：刘海龙

出版发行：清华大学出版社

网 址：<http://www.tup.com.cn>，<http://www.wqbook.com>

地 址：北京清华大学学研大厦 A 座 邮 编：100084

社 总 机：010-83470000 邮 购：010-62786544

投稿与读者服务：010-62776969，c-service@tup.tsinghua.edu.cn

质 量 反 馈：010-62772015，zhiliang@tup.tsinghua.edu.cn

印 装 者：北京同文印刷有限责任公司

经 销：全国新华书店

开 本：185mm×260mm 印 张：17.5 字 数：448 千字

版 次：2023 年 9 月第 1 版 印 次：2023 年 9 月第 1 次印刷

定 价：69.80 元

产品编号：100099-01

近年来，云计算技术在全球范围内迅猛发展，已经成为数字化时代的重要基础设施，深入到人们的生活和工作中。数字化、网络化、智能化、协同化已经成为世界经济和科技发展的主要趋势，我国正在逐步加强数字基础设施建设，积极发展新型基础设施，以进一步推进云计算、大数据、人工智能等新一代信息技术产业的发展。可以说，在当前数字化转型的背景下，云计算已经成为推动各行各业创新发展的重要动力。

KVM 虚拟化技术与 Docker 容器技术是云计算的两个重要“引擎”，它们分别在不同领域中具有独特的优势与适用场景。KVM 虚拟化是一种基于内核的虚拟机技术，在系统虚拟化方面有着广泛的应用。Docker 容器则是一个颠覆性的容器化技术，具有快速迁移、可移植性强等诸多优点。本书将围绕 KVM 虚拟化与 Docker 容器展开，探究它们的基本原理、适用场景、实现与管理等方面的知识，帮助读者更全面、更深入地了解这两项技术的特点以及如何使用它们来助力业务发展。

本书内容

本书共分 13 章，分为两部分：第一部分是 KVM 虚拟化技术(第 1~第 6 章)，第二部分是 Docker 容器技术(第 7~第 13 章)。

第 1 章介绍虚拟化的定义与历史，KVM 的原理、功能与优势。

第 2 章讲解 KVM 的架构，如何构建实验环境，如何在 Linux 宿主机中安装部署 KVM。

第 3 章讲解如何通过图形化管理工具 `virt-manager` 和命令行管理工具 `virt-install` 创建虚拟机。

第 4 章讲解如何通过 `virsh` 命令管理虚拟机，包括创建、暂停、恢复、停止及删除等生命周期管理的操作。

第 5 章讲解虚拟网络的管理，包括 NAT、桥接网络的原理与配置。

第 6 章讲解虚拟存储的管理，包括存储池、存储卷的原理与日常管理。

第 7 章介绍容器的定义、部署、发展历史及应用场景等。

第 8 章讲解 Docker 的架构原理、常用子命令、镜像及镜像仓库的使用。

第 9 章讲解 Docker 容器的基本概念及常用操作命令。

第 10 章讲解 Docker 网络原理，容器与外部网络的相互访问原理，默认网络类型分析，常用管理命令等。

第 11 章讲解 Docker 存储概述、存储卷的管理及适用场景。

第 12 章讲解 Dockerfile 的常用配置指令，如何使用 Dockerfile 创建镜像。

第 13 章讲解 Docker 实战案例：Linux 操作系统镜像、Nginx/Apache HTTPD 服务、MySQL、MariaDB、MongoDB 数据库服务。

适用对象

本书既可作为希望了解虚拟化技术和容器技术的系统管理员、DevOps 工程师的技术入门书籍，也可作为高等院校云计算相关课程的教材或教学参考书。在学习本书之前，应先具备一定的 Linux 基础知识。在学习过程中，既要充分理解相关的概念和原理，也要注重实践和动手操作，以加深对知识的理解和掌握。掌握 KVM 虚拟化技术与 Docker 容器技术，能够为后续学习 OpenStack、K8S 等云平台打下坚实基础。本书所有的实验及案例，均在 CentOS Stream 9 Linux 环境中经过验证。本书作为教材使用时，建议课时安排 64~72 学时。

本书由陈涛任主编，孔繁玉、孙鹏军、张炉焱任副主编，编写团队既有多年项目研发的历练，又有一线教学经验。本书编写过程中参考了国内外一些相关书籍，谨向这些作者表示诚挚的感谢。由于时间仓促，加之编者水平有限，书中难免存在不足之处，敬请广大读者批评指正。

本书 PPT 教学课件和案例源文件可通过扫描下方二维码下载。

服务邮箱：476371891@qq.com。



教学资源

编者

2023 年 3 月



C O N T E N T S

第 1 章 KVM 概述	1
1.1 虚拟化技术简介	1
1.1.1 虚拟化的定义	1
1.1.2 操作系统虚拟化的发展历史	2
1.1.3 虚拟化的分类	2
1.1.4 虚拟化的目的	3
1.2 KVM 简介	3
1.2.1 什么是 KVM	4
1.2.2 KVM 的发展史	4
1.2.3 KVM 历史大事件	4
1.2.4 KVM 的功能	4
1.2.5 KVM 的优势	5
1.3 习题	6
第 2 章 安装 KVM	7
2.1 KVM 架构解析	7
2.1.1 KVM 内核模块	7
2.1.2 QEMU 用户态工具	8
2.2 安装环境准备	8
2.2.1 Windows 环境准备	8
2.2.2 下载与安装 VMware Workstation	10
2.2.3 下载 CentOS Stream 9 的 ISO 文件	13
2.2.4 创建新虚拟机	14
2.3 图形化安装 KVM	20
2.3.1 克隆虚拟机	20
2.3.2 为虚拟机安装操作系统	23
2.3.3 图形化下安装 KVM	29
2.4 系统最小化安装 KVM	32

2.5 KVM 管理工具	37
2.5.1 libvirt	37
2.5.2 virsh	37
2.5.3 virt-manager	38
2.5.4 virt-viewer	38
2.6 习题	39
第 3 章 创建 KVM 虚拟机	41
3.1 Virt-Manager 图形化创建 KVM 虚拟机	41
3.1.1 创建虚拟机	41
3.1.2 使用 Virt-Manager 查看 当前配置	51
3.2 virt-install 命令创建虚拟机	52
3.2.1 创建虚拟机并通过交互 模式安装	53
3.2.2 查看虚拟机与环境的配置	55
3.2.3 virt-install 高级用法示例	55
3.3 VNC 连接 KVM 虚拟机	57
3.3.1 什么是 VNC	57
3.3.2 VNC 服务端	58
3.3.3 VNC 客户端	59
3.4 习题	61
第 4 章 虚拟机管理	63
4.1 libvirt 架构描述	63
4.2 使用 virsh 管理虚拟机	64
4.2.1 获得帮助	65
4.2.2 常用子命令	67
4.3 习题	74

第 5 章 管理 KVM 虚拟网络	75	6.6 习题	122
5.1 查看默认网络环境	75	第 7 章 容器技术简介	123
5.1.1 查看宿主机的网络环境	75	7.1 容器的定义	123
5.1.2 查看 libvirt 的网络环境	77	7.2 实验环境部署	124
5.1.3 查看虚拟机的网络配置	80	7.3 容器与虚拟机	126
5.1.4 libvirt 管理的虚拟网络	81	7.4 容器的发展史	129
5.2 创建和管理隔离网络	81	7.5 容器的标准化	131
5.2.1 通过 virsh 创建和管理隔离网络	82	7.6 容器的应用场景	132
5.2.2 使用隔离网络	84	7.7 习题	134
5.3 创建和管理 NAT 网络	85	第 8 章 Docker 镜像管理	135
5.3.1 使用 virsh 创建 NAT 网络	86	8.1 镜像的作用	135
5.3.2 使用 NAT 网络	89	8.2 获取镜像	136
5.4 创建和管理桥接网络	90	8.3 镜像的结构	138
5.4.1 在宿主机上创建网桥	91	8.4 Docker 的存储驱动程序	139
5.4.2 使用网桥	93	8.5 查看镜像信息	144
5.5 习题	94	8.5.1 使用 images/image ls 子命令列出镜像	145
第 6 章 管理 KVM 虚拟存储	95	8.5.2 使用 tag 子命令为镜像添加标签	146
6.1 常见的存储资源	95	8.5.3 使用 inspect 子命令查看详细的信息	148
6.2 虚拟磁盘类型	96	8.5.4 使用 history 子命令查看镜像的构建历史	148
6.3 qemu-img 磁盘管理命令	97	8.6 在 Docker 官方仓库中搜寻镜像	149
6.3.1 创建和格式化磁盘文件	97	8.7 删除和清理镜像	150
6.3.2 调整磁盘文件的大小	99	8.7.1 镜像的状态	150
6.3.3 镜像文件格式转换	100	8.7.2 删除镜像	151
6.3.4 快照管理	101	8.7.3 清理镜像	152
6.4 存储池	103	8.8 创建新镜像	153
6.4.1 查看当前存储池	104	8.8.1 基于已有容器创建新镜像	153
6.4.2 存储池分类	105	8.8.2 使用 Dockerfile 创建新镜像	156
6.4.3 创建基于目录的存储池	106	8.8.3 导入本地模板来创建新镜像	158
6.4.4 创建基于 LVM 逻辑卷的存储池	109	8.9 保存与加载镜像	160
6.4.5 创建基于网络文件系统的存储池	111	8.10 集中管理镜像	163
6.5 存储卷	114		
6.5.1 查看存储卷信息	115		
6.5.2 创建存储卷	116		
6.5.3 存储卷管理	120		

8.10.1	上传镜像到公共仓库	163	10.2.1	名称解析器默认的配置	196
8.10.2	上传镜像到私有仓库	165	10.2.2	修改解析器的配置	197
8.11	习题	168	10.3	容器的访问控制	199
第 9 章	Docker 容器管理	171	10.3.1	容器访问外部网络	199
9.1	容器管理概述	171	10.3.2	容器之间相互访问	203
9.2	创建容器	173	10.4	容器的端口映射	206
9.2.1	创建新容器	173	10.5	容器的便捷互联机制	208
9.2.2	启动容器	175	10.6	容器的网络管理命令	210
9.2.3	新建并启动容器	176	10.6.1	列出网络	210
9.2.4	在后台运行容器	177	10.6.2	查看网络信息	211
9.2.5	查看容器输出	178	10.6.3	创建自定义网络	213
9.3	停止容器	178	10.6.4	接入网络	218
9.3.1	暂停/恢复容器	178	10.6.5	断开网络	220
9.3.2	停止容器	180	10.6.6	删除和清理网络	222
9.3.3	杀死容器	181	10.7	配置 host 网络模式	223
9.4	进入容器内部	182	10.8	配置 none 网络模式	225
9.4.1	attach 子命令	182	10.9	习题	225
9.4.2	exec 子命令	183	第 11 章	Docker 存储管理	227
9.5	删除容器	184	11.1	Docker 存储概述	227
9.5.1	rm 子命令	184	11.2	Docker 的卷	228
9.5.2	prune 子命令	185	11.2.1	Docker 卷的管理	228
9.6	迁移容器	186	11.2.2	Docker 卷的使用	230
9.7	查看容器	187	11.2.3	Docker 卷的适用场景	231
9.7.1	查看容器详情	187	11.3	Docker 的绑定挂载	231
9.7.2	查看容器内进程	188	11.4	习题	234
9.7.3	查看统计信息	188	第 12 章	使用 Dockerfile 创建镜像	235
9.8	其他容器命令	189	12.1	Dockerfile 的基本结构	235
9.8.1	复制文件	189	12.2	Dockerfile 的配置指令	242
9.8.2	查看变更	189	12.3	Dockerfile 的操作指令	247
9.8.3	查看端口映射	190	12.4	创建镜像	250
9.8.4	更新配置	190	12.4.1	命令选项	250
9.9	习题	191	12.4.2	父镜像的选择	251
第 10 章	Docker 网络管理	193	12.4.3	使用 .dockerignore 文件	252
10.1	Docker 网络的启动和配置	193	12.4.4	多步骤创建	252
10.1.1	网络启动过程	193	12.5	习题	253
10.1.2	网络配置参数	195			
10.2	容器的名称解析	196			

第 13 章 Docker 实战案例	255
13.1 Linux 操作系统镜像	255
13.1.1 BusyBox	256
13.1.2 Alpine	257
13.1.3 Debian/Ubuntu	258
13.1.4 CentOS/Fedora	259
13.2 为镜像添加 SSH 服务	259
13.3 Web 服务	262
13.3.1 Nginx	262

13.3.2 Apache HTTPD	263
13.4 数据库服务	263
13.4.1 MySQL	264
13.4.2 MariaDB	266
13.4.3 MongoDB	266
13.5 习题	268
参考文献	269

KVM概述

现今人们已经越来越离不开互联网。在互联网上，云计算技术已经非常成熟。虚拟化是云计算的核心技术之一，也是云计算的底层支撑技术之一。利用虚拟化技术，可以实现 IT 资源弹性分配，使 IT 资源分配更加灵活，满足多样化的应用需求。现如今，KVM(kernel-based virtual machine, 基于内核的虚拟机)可以在大多数 Linux 发行版上运行，并且 KVM 也是很多公共云服务商的默认管理程序。



本章要点

- ◎ 虚拟化概述
- ◎ KVM 概述
- ◎ KVM 的优点



1.1 虚拟化技术简介

要学好 KVM，就需要掌握一些与虚拟化相关的基础知识，包括虚拟化的概念、发展史和实现方式。

1.1.1 虚拟化的定义

虚拟化是一个广义的术语，通常是指将计算机原件在虚拟的物理机上运行而不是在真实的物理机上运行。通俗来讲，就是可以将单 CPU 模拟成多个 CPU 并行，允许一个平台同时运行多个操作系统，并且应用程序都可以在互相独立的空间内运行而互不影响。

1.1.2 操作系统虚拟化的发展历史

操作系统虚拟化通常表现为在单一系统上运行多个虚拟操作系统，这些虚拟操作系统同时运行，而且又相互独立。

操作系统虚拟化技术的发展史如图 1-1 所示。从图 1-1 中可看出，20 世纪 60 年代就已经出现了虚拟化技术，最早是在 IBM CP-40 大型机上使用了虚拟内存和虚拟机的概念。

除此之外，IBM 公司还有很多与虚拟化有关的创新贡献。

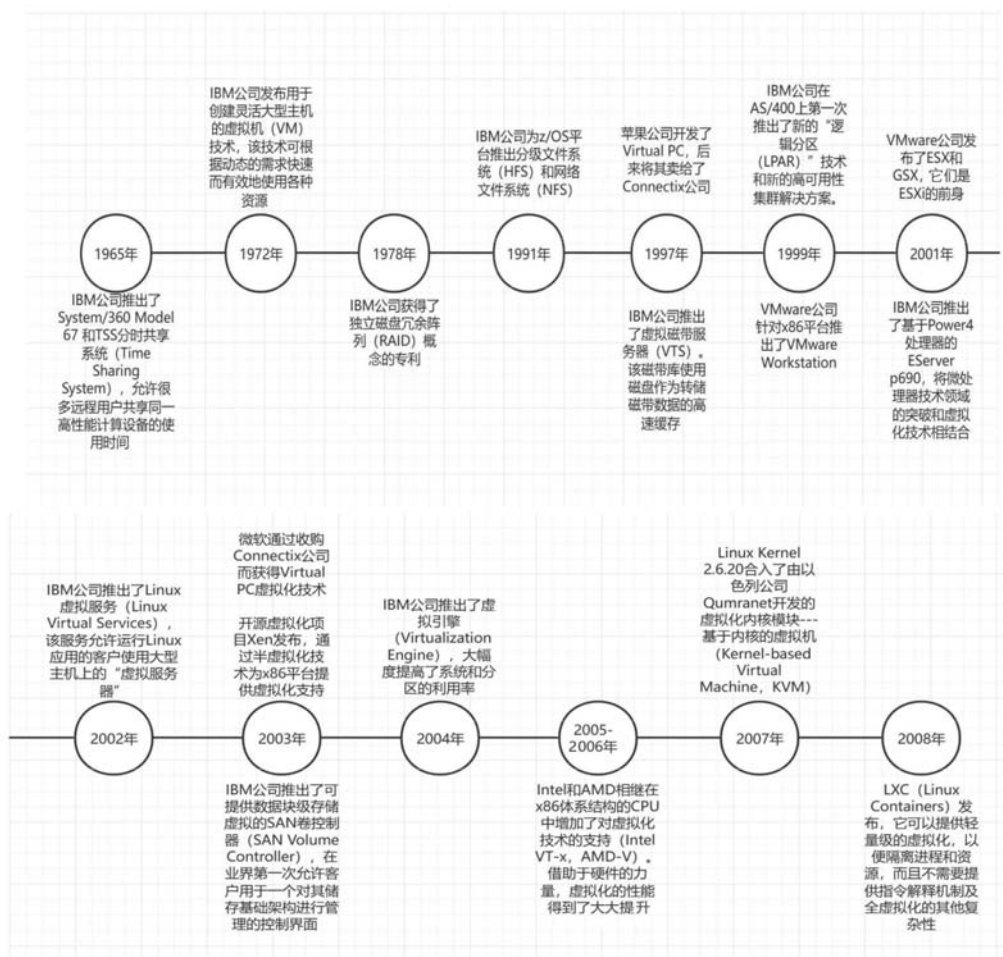


图 1-1

1.1.3 虚拟化的分类

虚拟化的方式多种多样，耳熟能详的名字有：全虚拟化、类虚拟化、硬件虚拟化、混合虚拟化等。这些不同的虚拟化方式，并不是根据同一个标准来分类的。

1. 根据虚拟化平台划分

从虚拟化平台的角度来划分，可以分为全虚拟化和类虚拟化。

(1) 全虚拟化：指虚拟机监视程序 (virtual machine monitor, VMM) 虚拟出来的平台是显示存

在的平台，客户机并不知道自己在运行在虚拟平台上，全虚拟化中的客户机操作系统是不需要做任何修改的。

(2) 类虚拟化：指通过对客户机进行源代码级的修改，让客户机可以使用虚拟化的资源。类虚拟化一般也被用来优化 I/O，客户机的操作系统通过高度优化的 I/O 协议，可以和 VMM 紧密结合达到近似于物理机的速度。

2. 根据应用对象划分

根据应用对象的不同，虚拟化技术可以分为以下几种。

(1) 操作系统虚拟化：将资源划分为一个或多个执行环境，以运行不同的操作系统。

(2) 存储虚拟化：对存储资源进行抽象化表现，从早期的 RAID、LVM 到后来的 VTL(虚拟化磁带库)，以及现在软件定义的存储等都是存储虚拟化。

(3) 网络虚拟化：软件定义的网络(software defined network, SDN)主要分为两个方向：控制平面虚拟化与数据平面虚拟化。

(4) GPU 虚拟化：将图形处理器单元(graphics processing unit, GPU)虚拟化之后，可以让运行在数据中心服务器上的虚拟机实例共享使用同一块或多块 GPU 处理器。

(5) 软件虚拟化：与常用的“绿色”软件类似，可以大大降低企业中软件部署、维护、升级的综合成本

(6) 硬件虚拟化：例如 SR-IOV(single root I/O virtualization)技术对网卡等设备进行封装、管理、共享，在其上创建多个 VF(virtual function)，每个 VF 对操作系统来讲都是真实的物理设备。

1.1.4 虚拟化的目的

虚拟化的主要目的是对 IT 基础设置进行简化，以及对资源进行访问。上面我们已经讲解过了虚拟化的原理，所以这里不再赘述。

虚拟化使用软件的方法重新定义及划分 IT 资源，可以实现 IT 资源的动态分配、灵活调度、跨域共享，提高 IT 资源利用率，使 IT 资源能够真正成为社会基础设施，服务于各行各业。

与传统 IT 资源分配的应用方式相比，虚拟化具有以下优势：

(1) 虚拟化技术可以大大提高资源的利用率，提供相互隔离、安全、高效的应用环境。

(2) 虚拟化系统能够方便地管理和升级资源。

虚拟化技术等的发展促进了云计算技术的飞速发展，也可以说虚拟化是云计算的基础，没有虚拟化就没有云计算。

1.2 KVM 简介

KVM既可以指KVM技术本身，也可以指KVM开源虚拟化解决方案。与VMware、Microsoft公司的商业虚拟化产品不同，KVM虚拟化解决方案是由KVM、QEMU和libvirt这3个既独立又合作的开源项目组合在一起形成的。

1.2.1 什么是 KVM

KVM(用于基于内核的虚拟机)是用于 x86 硬件上的 Linux 的完整虚拟化解决方案, 包含虚拟化扩展(Intel VT 或 AMD-V)。它由一个可加载的内核模块 `kvm.ko` 和一个处理器特定模块 `kvm-intel.ko` 或 `kvm-amd.ko` 组成, 该模块提供核心虚拟化基础设施。

使用 KVM, 可以运行多个未修改的 Linux 或 Windows 镜像的虚拟机。每个虚拟机都有私有的虚拟化硬件: 网卡、磁盘、图形适配器等。

KVM 是开源软件。从 KVM 2.6.20 开始, KVM 的内核组件包含在主线 Linux 中。从 KVM 1.3 开始, KVM 的用户空间组件包含在主线 QEMU 中。

由此可以看出, KVM 是基于内核的虚拟机。

1.2.2 KVM 的发展史

根据维基百科的记载:

(1) 阿维·齐维迪(Avi Kivity)在一家名为 Qumranet 的初创企业开始 KVM 的研发工作, 随后此公司于 2008 年被 Red Hat(红帽)公司收购。

(2) KVM 被合并入 Linux 内核版本 2.6.20 的主流分支, 于 2007 年 2 月 5 日发布。

(3) KVM 现由保罗·邦齐尼(Paolo Bonzini)维护。

1.2.3 KVM 历史大事件

2005 年 11 月, Intel 发布正式支持 VT 技术的处理器。

2006 年 5 月, AMD 发布支持 AMD-V 技术的处理器。

KVM 虚拟机最初由一个以色列的创业公司 Qumranet 开发, 作为它们的 VDI 产品的虚拟机。为了简化开发, KVM 的开发人员并没有选择从底层开始新写一个 Hypervisor, 而是选择了基于 Linux kernel, 通过加载新的模块从而使 Linux Kernel 本身变成一个 Hypervisor。

2006 年 8 月, 在先后完成了基本功能、动态迁移以及主要的性能优化之后, Qumranet 正式对外宣布 KVM 的诞生并推向 Linux 内核社区。同年 10 月, KVM 模块的源代码被正式接纳进入 Linux Kernel(内核), 成为内核源代码的一部分。

2007 年 2 月发布的 Linux 2.6.20 是第一个带有 KVM 模块的 Linux 内核正式发布版本。

2008 年 9 月 4 日, 同内核社区保持着很深渊源的著名 Linux 发行版提供商——Redhat 公司出人意料地收购 Qumranet, 从而成为 KVM 开源项目的新东家, 投入较多资源在 KVM 虚拟化开发中。

2010 年 11 月, Redhat 公司推出了新的企业版 Linux——RHEL 6, 在这个发行版本中集成了最新的 KVM 虚拟机, 而去掉了在 RHEL 5.x 系列中集成的 Xen。KVM 成为 RHEL 默认的虚拟化方案。

1.2.4 KVM 的功能

KVM 所支持的功能包括:

- 支持 CPU 和内存超额分配(overcommit)
- 支持半虚拟化 I/O(virtio)

- 支持热插拔(CPU、块设备、网络设备等)
- 支持对称多处理(symmetric multi-processing, SMP)
- 支持实时迁移(live migration)
- 支持 PCI 设备直接分配和单根 I/O 虚拟化(SR-IOV)
- 支持内核同页合并(kernel samepage merging, KSM)
- 支持 NUMA (non-uniform memory access, 非一致存储访问结构)

1.2.5 KVM 的优势

KVM 是 Linux 的一部分, Linux 也是 KVM 的一部分。Linux 有的优势, KVM 全都有。然而, KVM 的某些特点让它成为了企业的首选虚拟机监控程序。

1. 安全性

KVM 利用安全增强型 Linux(SELinux)和安全虚拟化(sVirt)组合来加强虚拟机的安全性和隔离性。SELinux 在虚拟机周围建立安全边界。sVirt 则扩展 SELinux 的功能, 使强制访问控制(MAC)安全机制应用到客户虚拟机, 并预防手动标记错误。

2. 存储方便

KVM 能够使用 Linux 支持的任何存储, 包括某些本地磁盘和网络附加存储(NAS)。还可以利用多路径 I/O 来增强存储并提供冗余能力。KVM 还支持共享文件系统, 因此虚拟机镜像可以由多个主机共享。磁盘镜像支持精简置备, 可以按需分配存储, 不必预先备妥一切。

3. 硬件支持

KVM 可以使用多种多样的认证 Linux 兼容硬件平台。由于硬件供应商经常助力内核开发, 所以 Linux 内核中通常能快速采用最新的硬件功能。

4. 内存管理

KVM 继承了 Linux 的内存管理功能, 包括非统一内存访问和内核同页合并。虚拟机的内存可以交换, 也可通过大型宗卷支持来提高性能, 还可由磁盘文件共享或支持。

5. 实时迁移

KVM 支持实时迁移, 也就是能够在物理主机之间移动运行中的虚拟机, 而不会造成服务中断。虚拟机保持开机状态, 网络连接保持活跃, 各个应用也会在虚拟机重新定位期间正常运行。KVM 也会保存虚拟机的当前状态, 从而存储下来供日后恢复。

6. 性能和可扩展性

KVM 继承了 Linux 的性能, 针对客户机和请求数量的增长进行扩展, 满足负载的需求。KVM 可让要求最苛刻的应用工作负载实现虚拟化, 而这也是许多企业虚拟化设置的基础, 如数据中心和私有云(基于 OpenStack)等。

7. 调度和资源控制

在 KVM 模型中, 虚拟机是一种 Linux 进程, 由内核进行调度和管理。通过 Linux 调度程

序，可对分配给 Linux 进程的资源进行精细控制，并且保障特定进程的服务质量。在 KVM 中，这包括完全公平的调度程序、控制组、网络命名空间和实时扩展。

8. 更低延迟，更高优先级

Linux 内核提供实时扩展，允许基于虚拟机的应用以更低的延迟、更高的优先级来运行(相对于裸机恢复)。内核也将需要长时间计算的进程划分为更小的组件，再进行相应的调度和处理。

1.3 习题

1. 用自己的话描述什么是虚拟化。
2. 用自己的话描述什么是 KVM。
3. KVM 有哪些特点？