



KVM in Action

Theory, Advanced Practice and Performance Tuning

KVM实战

原理、进阶与性能调优

任永杰 程舟 著

阿里云和Intel的云计算与虚拟化技术专家联合撰写

系统性、实战性兼备的KVM学习手册，从基本原理、实战进阶和性能优化三个维度详细讲解KVM的各种技术细节



机械工业出版社
China Machine Press

云计算与虚拟化技术丛书



KVM in Action

Theory, Advanced Practice and Performance Tuning

KVM实战

原理、进阶与性能调优

任永杰 程舟 著



机械工业出版社
China Machine Press

图书在版编目 (CIP) 数据

KVM 实战：原理、进阶与性能调优 / 任永杰，程舟著. —北京：机械工业出版社，2019.2
(云计算与虚拟化技术丛书)

ISBN 978-7-111-61981-9

I. K… II. ①任… ②程… III. 虚拟处理机 IV. TP338

中国版本图书馆 CIP 数据核字 (2019) 第 025068 号

KVM 实战：原理、进阶与性能调优

出版发行：机械工业出版社（北京市西城区百万庄大街 22 号 邮政编码：100037）

责任编辑：孙海亮

责任校对：李秋荣

印 刷：北京市兆成印刷有限责任公司

版 次：2019 年 3 月第 1 版第 1 次印刷

开 本：186mm×240mm 1/16

印 张：28.5

书 号：ISBN 978-7-111-61981-9

定 价：89.00 元

凡购本书，如有缺页、倒页、脱页，由本社发行部调换

客服热线：(010) 88379426 88361066

投稿热线：(010) 88379604

购书热线：(010) 68326294 88379649 68995259

读者信箱：hzit@hzbook.com

版权所有·侵权必究

封底无防伪标均为盗版

本书法律顾问：北京大成律师事务所 韩光 / 邹晓东

为什么要写这本书

自《KVM 虚拟化技术：实战与原理解析》（以下简称“上一本书”）出版以来，受到了读者的热烈欢迎，几度脱销重印。这给了笔者强烈的鼓舞和责任感，觉得有必要与时俱进给读者介绍最新的 KVM 虚拟化技术的相关知识。

从上一本书出版后到现在近 5 年时间里，国内虚拟化技术迅速普及，云计算应用风起云涌，阿里云、腾讯云、华为云等国内云服务提供商迅速崛起，使得云计算、虚拟化不再是原来象牙塔里虚无缥缈的技术概念，而是与普通大众日常生活息息相关的新名词，KVM 被这几大云服务提供商广泛采用^①，使得它成为云计算世界里事实上的虚拟化标准。在这样的市场背景下，以 Intel 为代表的 x86 硬件厂商，这些年也愈加重视虚拟化技术的硬件支持与创新^②，ARM 平台的硬件虚拟化支持也愈加完善。硬件层面的创新也促使 QEMU、KVM 在软件层面日新月异。比如，从 2013 年第 1 版发行至今，KVM（内核）版本从 3.5 发展到了 4.8；QEMU 版本从 1.3 发展到了 2.7；专门针对 KVM 的 qemu-kvm 代码树已经废弃（被合并到了主流 QEMU 中）……因此，上一本书中的很多用例、方法和结论等，在新的代码环境下已经有些不合时宜，甚至会出错。我们有必要给读者提供最新且正确的信息。

相对于上一本书的修改

总体来说，我们对上一本书里所有的用例、图例都做了大量修改更新，实验环境采用笔者写作时的最新技术：硬件平台采用 Intel Broadwell Xeon Server，KVM（内核）为 4.8 版本，QEMU 为 2.7 版本，操作系统环境是 RHEL 7.3。文中注释改成脚注的形式，而不是

① 全球最大的云服务提供商 Amazon，在 2017 年 11 月也宣布开始采用 KVM 作为它的虚拟化技术，大有启用 KVM 而放弃 Xen 之势。

② 硬件虚拟化支持是每一代 Intel CPU 的最核心功能之一，也是每一代新平台新功能和性能提升的领域。

像上一本书那样出现在每章末尾。我们认为这样更方便读者阅读。当然，文字表述上也进行了许多修改。

另外，我们对章节的结构也进行了重新组织，全书共分3篇，10章：第一篇“KVM 虚拟化基础”（第1章～第5章），第二篇“KVM 虚拟化进阶”（第6章～第9章），第三篇“性能测试与调优”（第10章）。

除了上述的总体修改外，各章主要修改内容如下。

第1章，我们重新组织了结构，精简了一些文字介绍，加入了一些数据图表以便于读者的理解。加入了云计算几种服务模型的描述和图示。加入了一节关于容器（Container）的简介，以便读者对比学习。

第2章，对上一本书相关章节进行了更为系统的梳理，介绍了硬件虚拟化技术、KVM、QEMU、与KVM配合的组件以及相关工具链。

第5章，着重对其中的网络配置一节进行了更新。将上一本书中的第5章拆分成了第6章、第7章、第8章和第9章，并分别进行了内容扩充。

第6章，对应上一本书中的第5章的半虚拟化、设备直接分配、热插拔这3节，并分别进行了补充。在半虚拟化驱动一节中，我们新增了“内核态的 vhost-net 后端以及网卡多队列”“使用用户态的 vhost-user 作为后端驱动”“对 Windows 客户机的优化”这3小节。在设备直接分配一节中，我们使用 VFIO 替换掉了已经被废弃的 Legacy passthrough。在热插拔一节，我们将内存热插拔独立出来，并着重更新，因为在上一本书出版时它还未被完全支持。除此之外，我们还新增了磁盘热插拔和网络接口的热插拔两节。

第7章，我们将上一本书中的第4章中内存大页部分和上一本书中的第5章中的 KSM、透明大页等内容凑在一起，组成了 KVM 内存管理高级技巧，同时新增了 NUMA（非统一内存访问架构）一节。

第8章，由上一本书中的第5章的“动态迁移”和“迁移到 KVM 虚拟化环境”两节组成。

第9章，在上一本书中的第5章的“嵌套虚拟化”“KVM 安全”等内容的基础上，新增了“CPU 指令相关的性能优化”一节，着重介绍了最近几年 Intel 的一些性能优化新指令在虚拟化环境中的应用。

第10章，对应上一本书中的第8章，专门讲 KVM 性能测试与优化。我们在最新的软硬件环境中重做了 CPU、内存、网络、磁盘的性能测试，获取了最新的数据，尤其对一些测试工具（benchmark）进行了重新选取，比如磁盘性能测试，我们放弃了 IOzone 和 Bonnie++，而选用业界更认可的 fio。另外，我们还加入了“CPU 指令集对性能的提升”和“其他的影响客户机性能的因素”两节进行分析，希望对读者进行虚拟化系统调优有所启示。

上一本书中的第7章“Linux 发行版中的 KVM”和第9章“参与 KVM 开源社区”分别作为本书的附录 A 和附录 B，并进行了相应的内容更新。

其他章节的内容保持不变，即第3章为上一本书的第3章且内容不变；第4章为上一本书的第6章，内容不变。

读者对象

本书适合对 Linux 下虚拟化或云计算基础技术感兴趣的读者阅读，包括 Linux 运维工程师、KVM 开发者、云平台开发者、虚拟化方案决策者、KVM 的用户以及其他对 KVM 虚拟机感兴趣的计算机爱好者。希望本书对这些读者了解 KVM 提供以下帮助。

- ❑ Linux 运维工程师：了解 KVM 的使用方法、功能和基本的性能数据，能够搭建高性能的 KVM 虚拟化系统，并应用于生产环境中。
- ❑ KVM 开发者：了解 KVM 的基本原理和功能，也了解其基本用法和一些调试方法，以及如何参与到 KVM 开源社区中去贡献代码。
- ❑ 云平台开发者：了解底层 KVM 虚拟化的基本原理和用法，以促进云平台上层应用的开发和调试的效率。
- ❑ 虚拟化方案决策者：了解 KVM 的硬件环境需求和它的功能、性能概况，以便在虚拟化技术选型时做出最优化的决策。
- ❑ 普通用户：了解 KVM 的功能和如何使用 KVM，用掌握的 KVM 虚拟化技术来促进其他相关的学习、开发和测试。

如何阅读本书

前面已经提到，本书相比上一本书内容更加集中，分类更加合理。如果读者朋友对 KVM 没有什么了解，笔者建议按本书章节顺序阅读，通读一遍之后再对感兴趣的章节进行仔细阅读。对于已有一定 KVM 知识基础的读者，可以根据自己的兴趣和已经掌握的知识情况来有选择地阅读各个章节。当然，笔者建议今后可能会经常使用 KVM 的读者，在阅读本书时，可以根据书中示例或者其他示例来进行实际操作。如果是开发者，也可以查看相应的源代码。

勘误和支持

KVM、QEMU 等开源社区非常活跃，QEMU/KVM 发展迅速，每天都有新的功能加进去，或者原有功能被改进。特别是 qemu 命令行参数很可能会有改动，故本书中 qemu 命令行参数只能完全适用于本书中提及的 QEMU 版本，读者若使用不同的版本，命令行参数可能并不完全相同。例如，本书写作时，“-enable-kvm”已经被社区标为“将要废弃”，很可能在读者拿到本书的时候，需要用“-accel kvm”来代替它。

由于 KVM 和 QEMU 的发展变化比较快，加之笔者的技术水平有限，编写时间仓促，书中难免会出现一些错误或者不准确的地方，恳请读者朋友批评指正。读者朋友对本书相关内容有任何的疑问、批评和建议，都可以通过笔者之一（任永杰）的博客网站 <http://>

smilejay.com/ 进行讨论。也可以发邮件给我们 (smile665@gmail.com、Robert.Ho@outlook.com), 笔者会尽力回复并给读者以满意的答案。全书中涉及的示例代码程序 (不包含单行的命令) 和重要的配置文件, 都可以从网站 <https://github.com/smilejay/kvm-book> 查看和下载。

如果读者朋友们有更多的宝贵意见或者任何关于 KVM 虚拟化技术的讨论, 也都欢迎发送电子邮件至邮箱 smile665@gmail.com、Robert.Ho@outlook.com, 我们非常期待能够得到朋友们的真挚反馈。

Contents 目 录

前言

第一篇 KVM 虚拟化基础

第 1 章 虚拟化简介..... 2

1.1 云计算概述..... 2

1.1.1 什么是云计算..... 2

1.1.2 云计算的历史..... 4

1.1.3 云计算的几种服务模型..... 5

1.2 虚拟化技术..... 6

1.2.1 什么是虚拟化..... 6

1.2.2 软件虚拟化和硬件虚拟化..... 7

1.2.3 半虚拟化和全虚拟化..... 8

1.2.4 Type1 和 Type2 虚拟化..... 8

1.3 KVM 简介..... 9

1.3.1 KVM 的历史..... 9

1.3.2 KVM 的功能概览..... 10

1.3.3 KVM 的现状..... 12

1.3.4 KVM 的展望..... 13

1.4 其他的虚拟化解决方案简介..... 14

1.4.1 Xen..... 14

1.4.2 VMware..... 14

1.4.3 HyperV..... 16

1.4.4 Container..... 17

1.5 本章小结..... 17

第 2 章 KVM 原理简介..... 18

2.1 硬件虚拟化技术..... 18

2.1.1 CPU 虚拟化..... 18

2.1.2 内存虚拟化..... 20

2.1.3 I/O 虚拟化..... 22

2.1.4 Intel 虚拟化技术发展..... 24

2.2 KVM 架构概述..... 25

2.3 KVM 内核模块..... 26

2.4 QEMU 用户态设备模拟..... 27

2.5 与 QEMU/KVM 结合的组件..... 28

2.6 KVM 上层管理工具..... 30

2.7 本章小结..... 31

第 3 章 构建 KVM 环境..... 32

3.1 硬件系统的配置..... 32

3.2 安装宿主机 Linux 系统..... 34

3.3 编译和安装 KVM..... 37

3.3.1 下载 KVM 源代码..... 37

3.3.2 配置 KVM	39	4.4.3 virt-top	107
3.3.3 编译 KVM	43	4.4.4 libguestfs	107
3.3.4 安装 KVM	44	4.5 云计算管理平台	109
3.4 编译和安装 QEMU	46	4.5.1 OpenStack 简介	109
3.4.1 曾经的 qemu-kvm	46	4.5.2 ZStack 简介	111
3.4.2 下载 QEMU 源代码	47	4.6 本章小结	111
3.4.3 配置和编译 QEMU	47		
3.4.4 安装 QEMU	49	第 5 章 KVM 核心基础功能	112
3.5 安装客户机	50	5.1 硬件平台和软件版本说明	112
3.6 启动第一个 KVM 客户机	53	5.2 CPU 配置	114
3.7 本章小结	53	5.2.1 vCPU 的概念	115
第 4 章 KVM 管理工具	55	5.2.2 SMP 的支持	116
4.1 libvirt	55	5.2.3 CPU 过载使用	123
4.1.1 libvirt 简介	55	5.2.4 CPU 模型	124
4.1.2 libvirt 的安装与配置	58	5.2.5 进程的处理亲和性和 vCPU 的 绑定	127
4.1.3 libvirt 域的 XML 配置文件	63	5.3 内存配置	131
4.1.4 libvirt API 简介	76	5.3.1 内存设置基本参数	132
4.1.5 建立到 Hypervisor 的连接	78	5.3.2 EPT 和 VPID 简介	133
4.1.6 libvirt API 使用示例	82	5.3.3 内存过载使用	136
4.2 virsh	87	5.4 存储配置	137
4.2.1 virsh 简介	87	5.4.1 存储配置和启动顺序	137
4.2.2 virsh 常用命令	88	5.4.2 qemu-img 命令	142
4.3 virt-manager	95	5.4.3 QEMU 支持的镜像文件格式	145
4.3.1 virt-manager 简介	95	5.4.4 客户机存储方式	149
4.3.2 virt-manager 编译和安装	96	5.5 网络配置	150
4.3.3 virt-manager 使用	96	5.5.1 用 QEMU 实现的网络模式	151
4.4 virt-viewer、virt-install、virt-top 和 libguestfs	105	5.5.2 使用直接的网桥模式	154
4.4.1 virt-viewer	105	5.5.3 用网桥实现 NAT 模式	161
4.4.2 virt-install	106	5.5.4 QEMU 内部的用户模式网络	170
		5.5.5 其他网络选项	174

5.6 图形显示	175
5.6.1 SDL 的使用	175
5.6.2 VNC 的使用	177
5.6.3 VNC 显示中的鼠标偏移	186
5.6.4 非图形模式	187
5.6.5 显示相关的其他选项	189
5.7 本章小结	191

第二篇 KVM 虚拟化进阶

第 6 章 KVM 设备高级管理

6.1 半虚拟化驱动	194
6.1.1 virtio 概述	194
6.1.2 安装 virtio 驱动	196
6.1.3 使用 virtio_balloon	203
6.1.4 使用 virtio_net	208
6.1.5 使用 virtio_blk	210
6.1.6 内核态的 vhost-net 后端以及 网卡多队列	212
6.1.7 使用用户态的 vhost-user 作为 后端驱动	215
6.1.8 kvm_clock 配置	216
6.1.9 对 Windows 客户机的优化	218
6.2 设备直接分配 (VT-d)	219
6.2.1 VT-d 概述	219
6.2.2 VFIO 简介	220
6.2.3 VT-d 环境配置	221
6.2.4 VT-d 操作示例	230
6.2.5 SR-IOV 技术	240
6.3 热插拔	251

6.3.1 PCI 设备热插拔	252
6.3.2 PCI 设备热插拔示例	253
6.3.3 CPU 的热插拔	259
6.3.4 内存的热插拔	260
6.3.5 磁盘的热插拔	264
6.3.6 网卡接口的热插拔	265
6.4 本章小结	266

第 7 章 KVM 内存管理高级技巧

7.1 大页	268
7.1.1 大页的介绍	268
7.1.2 KVM 虚拟化对大页的利用	271
7.2 透明大页	274
7.3 KSM	277
7.3.1 KSM 基本原理	278
7.3.2 KSM 操作实践	280
7.3.3 QEMU 对 KSM 的控制	286
7.4 与 NUMA 相关的工具	288
7.4.1 numastat	289
7.4.2 numad	290
7.4.3 numactl	294
7.5 本章小结	295

第 8 章 KVM 迁移

8.1 动态迁移	297
8.1.1 动态迁移的概念	297
8.1.2 动态迁移的效率和应用场景	298
8.1.3 KVM 动态迁移原理	299
8.1.4 KVM 动态迁移实践	302
8.1.5 VT-d/SR-IOV 的动态迁移	306
8.2 迁移到 KVM 虚拟化环境	307

8.2.1 virt-v2v 工具介绍.....	307	第三篇 性能测试与调优	
8.2.2 从 Xen 迁移到 KVM	308		
8.2.3 从 VMware 迁移到 KVM	310	第 10 章 KVM 性能测试及参考数据...374	
8.2.4 从 VirtualBox 迁移到 KVM	311	10.1 虚拟化性能测试简介	374
8.2.5 从物理机迁移到 KVM 虚拟化 环境 (P2V)	312	10.2 CPU 性能测试	376
8.3 本章小结	313	10.2.1 CPU 性能测试工具	376
第 9 章 其他高级功能	314	10.2.2 测试环境配置	378
9.1 嵌套虚拟化	314	10.2.3 性能测试方法	381
9.1.1 嵌套虚拟化的基本概念	314	10.2.4 性能测试数据	382
9.1.2 KVM 嵌套 KVM	315	10.3 内存性能测试	384
9.2 KVM 安全	318	10.3.1 内存性能测试工具	384
9.2.1 SMEP/SMAP/MPX	318	10.3.2 测试环境配置	385
9.2.2 控制客户机的资源使用—— cgroups	319	10.3.3 性能测试方法	385
9.2.3 SELinux 和 sVirt	327	10.3.4 性能测试数据	387
9.2.4 其他安全策略	336	10.4 网络性能测试	388
9.3 CPU 指令相关的性能优化	340	10.4.1 网络性能测试工具	388
9.3.1 AVX/AVX2/AVX512	340	10.4.2 测试环境配置	389
9.3.2 XSAVE 指令集	342	10.4.3 性能测试方法	391
9.3.3 AES 新指令	343	10.4.4 性能测试数据	393
9.3.4 完全暴露宿主机 CPU 特性	349	10.5 磁盘 I/O 性能测试	394
9.4 QEMU 监控器	351	10.5.1 磁盘 I/O 性能测试工具	394
9.4.1 QEMU monitor 的切换和配置	351	10.5.2 测试环境配置	395
9.4.2 常用命令介绍	352	10.5.3 性能测试方法	397
9.5 qemu 命令行参数	358	10.5.4 性能测试数据	400
9.5.1 回顾已用过的参数	358	10.6 CPU 指令集对性能的提升	403
9.5.2 其他常用参数	363	10.7 其他影响客户机性能的因素	405
9.6 本章小结	372	10.8 本章小结	407
		附录 A Linux 发行版中的 KVM	408
		附录 B 参与 KVM 开源社区	417

第一篇 *Part 1*

KVM 虚拟化基础

- 第 1 章 虚拟化简介
 - 第 2 章 KVM 原理简介
 - 第 3 章 构建 KVM 环境
 - 第 4 章 KVM 管理工具
 - 第 5 章 KVM 核心基础功能
- 

虚拟化简介

在写作上一本书的时候（2013 年），云计算虽然已经在国际上提出多年，但在国内还是刚刚兴起。到写作本书时这短短 4 年内，国内云计算已经翘首追赶，紧跟国际的步伐。例如阿里云，截至笔者写作时，阿里云已经连续 7 个季度 3 位数的同比增长率。老牌公司微软也在这几年加速追赶，在这个新的领域跻身一线。

1.1 云计算概述

1.1.1 什么是云计算

一直以来，云计算（Cloud Computing）的定义也如同它的名字一样，云里雾里，说不清楚。维基百科里是这样定义的：是一种基于互联网的计算方式，通过这种方式，共享的软硬件资源和信息可以按需求提供给计算机各种终端和其他设备[⊖]。以前，我们的信息处理（计算）是由一个实实在在的计算机来完成的，它看得见，摸得着。后来，随着计算硬件、网络技术、存储技术的飞速发展，人们发现，每个人独自拥有一台计算机似乎有些浪费，因为它大多数时候是空闲的。那么，如果将计算资源集中起来，大家共享，类似现代操作系统那样分时复用，将是对资源的极大节省和效率的极大提升，经济学上的解释也就是边际效应（成本）递减。科技行业的发展，根源也是经济利益的推动。在这样的背景下，云计算应运而生了。它就是把庞大的计算资源集中在某个地方或是某些地方，而不再是放在身边的那台计算机了；用户的每一次计算，都发生在那个被称为云的他看不见摸不着的某个地方。

以 CPU 为例，图 1-1 和图 1-2 摘选了从 2000 年到 2017 年上市的 Intel 桌面 CPU 的参数

⊖ <https://zh.wikipedia.org/wiki/%E9%9B%B2%E7%AB%AF%E9%81%8B%E7%AE%97>。

(主频、核数、LLC^①、制造工艺)，从中可以大概看到 CPU 处理能力的飞速提升。它的另一面也就意味着，个人单独拥有一台计算机，从资源利用效率角度来看，被大大闲置了的。

2000—2017年Intel桌面CPU主频及核数

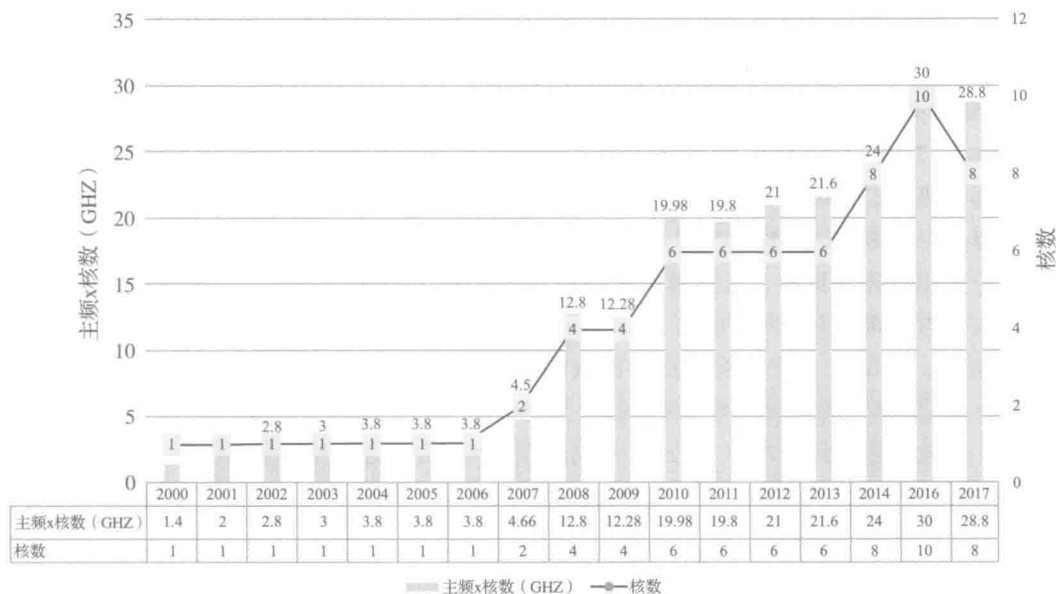


图 1-1 Intel CPU 处理能力的主要参数

2000—2017年Intel 桌面CPU LLC及制程

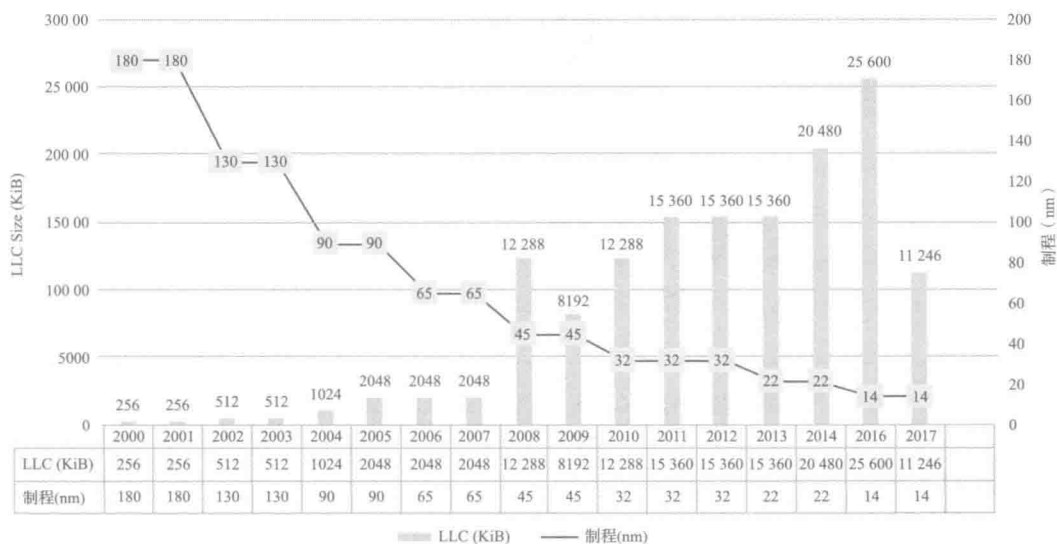


图 1-2 Intel CPU LLC 的发展及制造工艺的演进

① Last Level Cache 指介于 CPU core 和 RAM 之间的最后一层缓存。Pentium 时代通常最多到 L2 Cache。现在的 Core i3/i5/i7 都有 L3 Cache，并且越来越大。

1.1.2 云计算的历史

正式的云计算的产品始发于 2006 年，那年 8 月，亚马逊（Amazon）发布了“弹性计算云”（Elastic Compute Cloud）。2008 年 10 月，微软宣布了名为 Azure 的云计算产品，并在 2010 年 2 月正式发布 Windows Azure^①。Google 也从 2008 年开始进入云计算时代，那年 4 月，其发布了 Google App Engine Beta，但直到 2013 年 12 月，其 Google Compute Engine 对标 AWS EC2 才正式可用。

2010 年 7 月，NASA 和 Rackspace 共同发布了著名的开源项目 Openstack。

从国内来看，2009 年，阿里巴巴率先成立了阿里云部门，一开始只对内服务于其自身的电商业务，如淘宝、天猫。2011 年 7 月，阿里云开始正式对外销售云服务。

无论是国内还是国外，云计算的市场都快速发展。“Amazon 把云计算做成一个大生意没有花太长的时间：不到两年时间，Amazon 上的注册开发人员达 44 万人，还有为数众多的企业级用户。有第三方统计机构提供的数据显示，Amazon 与云计算相关的业务收入已达 1 亿美元。云计算是 Amazon 增长最快的业务之一。”^②国内的阿里云也在 2013 年以后快速蓬勃发展。无论是国际巨头 AWS（亚马逊）还是国内的阿里云，这些年都是快速增长，尤其阿里云，虽然体量暂时还远不及 AWS，但一直都是超过 100% 的增长，让人侧目，如图 1-3 所示。

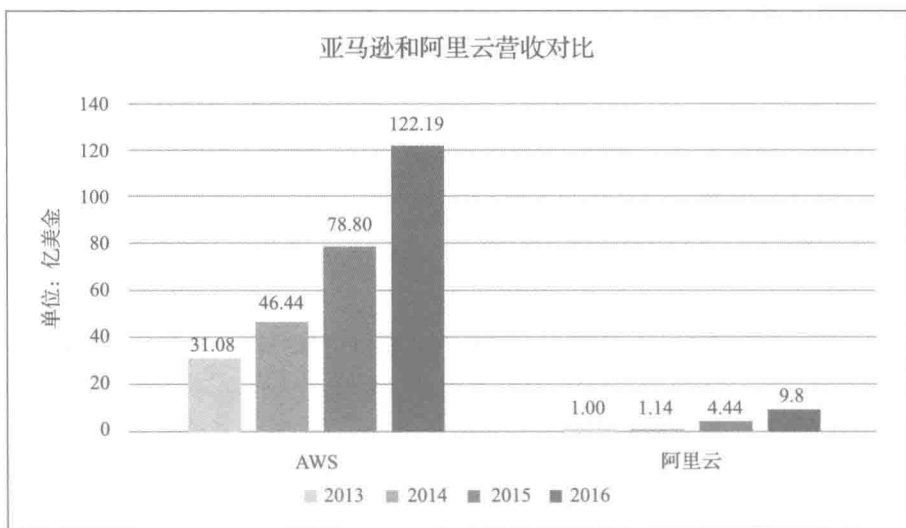


图 1-3 云计算营收对比：2013—2017 年度亚马逊 AWS 和阿里云^③

① 2014 年 3 月，Windows Azure 更名为 Microsoft Azure。

② 刘鹏：云计算发展现状。<http://www.chinacloud.cn/show.aspx?id=754&cid=11>。

③ 数据整理自网络。阿里财年始于前一年 4 月，这里以大约自然年表示，比如 2015 年数据来自于阿里 2016 财报。另外阿里云 2013、2014 年度数据包括其 internet infrastructure 收入，2015 年开始才将阿里云单独出来。换算时，美元兑人民币汇率假设为 6.8。微软的 Azure 营收没有单独披露。Google cloud 营收也未能单独获得。

1.1.3 云计算的几种服务模型

“云计算是推动 IT 转向以业务为中心模式的一次重大变革。它着眼于运营效率、竞争力和快速响应等实际成果。这意味着 IT 的作用正在从提供 IT 服务逐步过渡到根据业务需求优化服务的交付和使用。这种全新的模式将以前的信息孤岛转化为灵活高效的资源池和具备自我管理能力的虚拟基础架构，从而以更低的成本和以服务的形式提供给用户。IT 即服务将提供业务所需要的一切，并在不丧失对系统的控制力的同时，保持系统的灵活性和敏捷性。”^①

云计算的模型是以服务为导向的，根据提供的服务层次不同，可分为：IaaS (Infrastructure as a Service，基础架构即服务)、PaaS (Platform as a Service，平台即服务)、SaaS (Software as a Service，软件即服务)。它们提供的服务越来越抽象，用户实际控制的范围也越来越小，如图 1-4 所示。

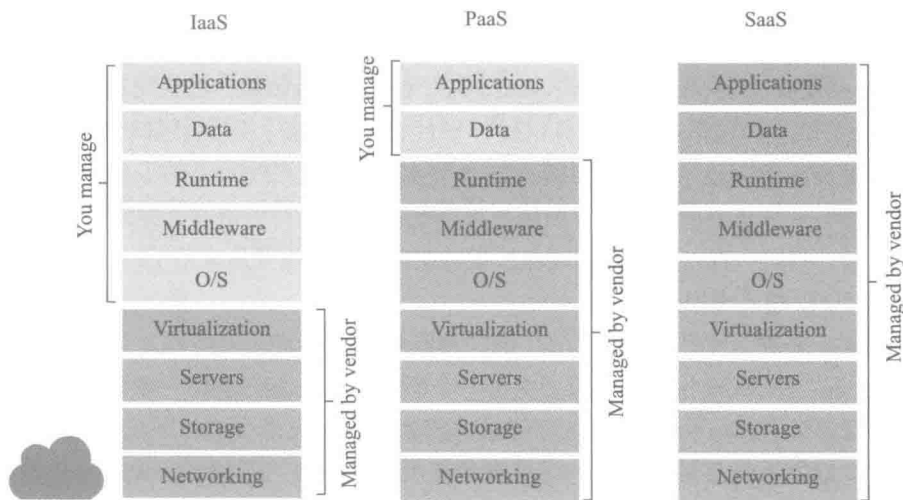


图 1-4 云计算服务层次模型^②

1. SaaS，软件即服务

云服务提供商提供给客户直接使用软件服务，如 Google Docs、Microsoft CRM、Salesforce.com 等。用户不必自己维护软件本身，只管使用软件提供的服务。用户为该软件提供的服务付费。

2. PaaS，平台即服务

云服务提供商提供给客户开发、运维应用程序的运行环境，用户负责维护自己的应用程序，但并不掌控操作系统、硬件以及运作的网络基础架构。如 Google App Engine 等。平台是指应用程序运行环境（图 1-4 中的 Runtime）。通常，这类用户在云环境中运维的应用程序会再提供软件服务给他的下级客户。用户为自己的程序的运行环境付费。

^① 虚拟化和云计算技术回顾与展望。http://www.ctosay.cn/content/40797008418208146787.html。

^② 图片来源 http://inspira.co.in/blog/benefits-iaas-vs-paas-vs-saas/。

3. IaaS，基础设施即服务

用户有更大的自主权，能控制自己的操作系统、网络连接（虚拟的）、硬件（虚拟的）环境等，云服务提供商提供的是一个虚拟的主机环境。如 Google Compute Engine、AWS EC2 等。用户为一个主机环境付费。

从图 1-4 中可以看到，无论是哪种云计算服务模型，虚拟化（Virtualization）都是其基础。那么什么是虚拟化呢？

1.2 虚拟化技术

1.2.1 什么是虚拟化

维基百科关于虚拟化的定义是：“In computing, virtualization refers to the act of creating a virtual (rather than actual) version of something, including virtual computer hardware platforms, storage devices, and computer network resources。”（在计算机领域，虚拟化指创建某事物的虚拟（而非实际）版本，包括虚拟的计算机硬件平台、存储设备，以及计算机网络资源）可见，虚拟化是一种资源管理技术，它将计算机的各种实体资源（CPU、内存、存储、网络等）予以抽象和转化出来，并提供分割、重新组合，以达到最大化利用物理资源的目的。

广义来说，我们一直以来对物理硬盘所做的逻辑分区，以及后来的 LVM（Logical Volume Manager），都可以纳入虚拟化的范畴。

结合图 1-4 来看，在没有虚拟化以前（我们抽掉 Virtualization 层），一个物理的主机（Sever、Storage、Network 层）上面只能支持一个操作系统及其之上的一系列运行环境和应用程序；有了虚拟化技术，一个物理主机可以被抽象、分割成多个虚拟的逻辑意义上的主机，向上支撑多个操作系统及其之上的运行环境和应用程序，则其资源可以被最大化地利用。

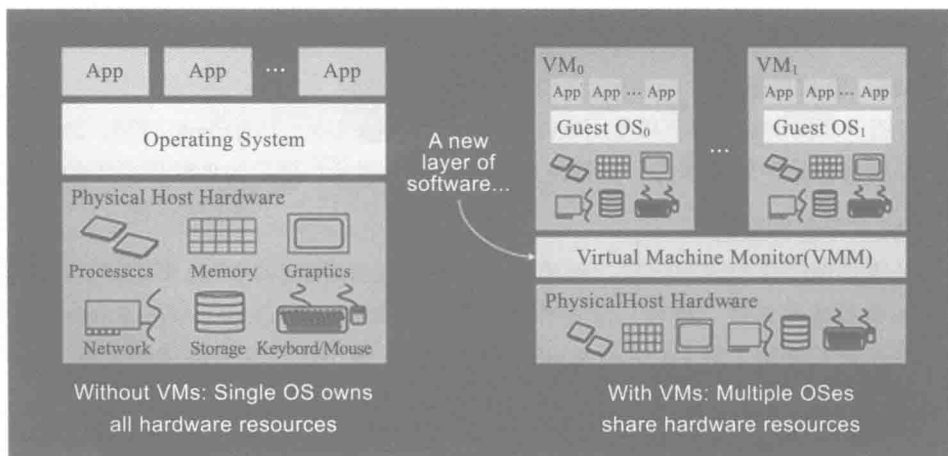


图 1-5 物理资源虚拟化示意