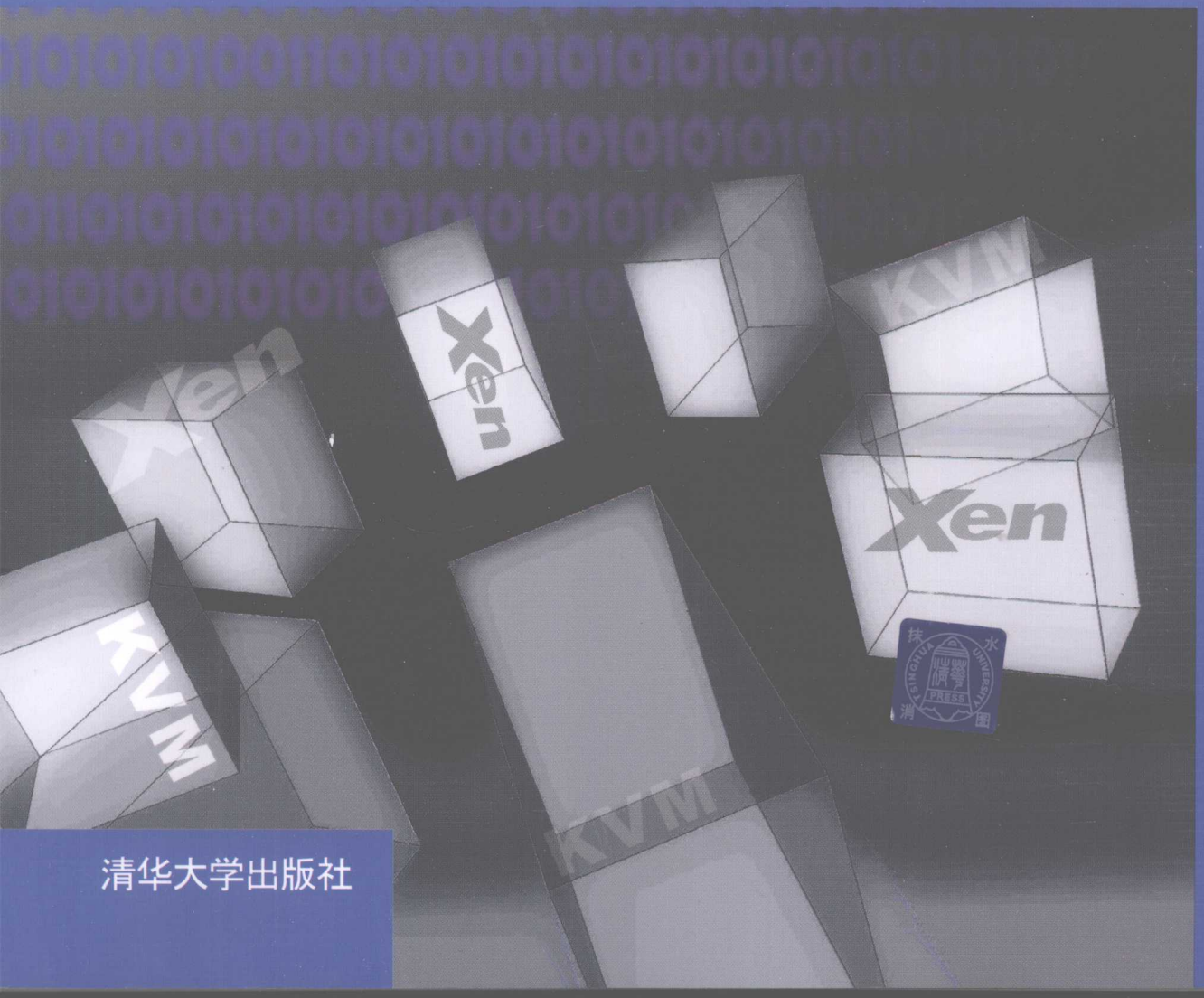
Intel® Corporation 2008
英特尔开源软件技术中心 著
复旦大学并行处理研究所

系统虚拟化
——原理与实现



清华大学出版社

系统虚拟化
——原理与实现

Intel® Corporation 2008
英特尔开源软件技术中心　　著
复旦大学并行处理研究所

清华大学出版社
北京

内 容 简 介

本书深入而又系统地介绍了以软件完全虚拟化、硬件辅助虚拟化及类虚拟化为核心的各种系统虚拟化技术。全书共9章，第1章概述性地介绍了虚拟化技术；第2章介绍计算机系统知识；第3章从CPU虚拟化、内存虚拟化和I/O虚拟化三大块对系统虚拟化技术进行概述，并介绍虚拟机监控器(VMM)的组成与分类，而且对市场上流行的虚拟化产品进行了简单介绍；第4～6章分别从基于软件的完全虚拟化、硬件辅助的完全虚拟化和类虚拟化三种实现技术角度深入介绍系统虚拟化方法；第7章介绍虚拟机的性能评测和调试技术；第8章介绍系统虚拟化的应用实例；最后在第9章对虚拟机和系统虚拟化技术的发展作一个展望。

本书是系统虚拟化技术实现原理的全面展示，也是作者这些年在虚拟化学术和工业研究领域开发的经验总结。本书理论与实践相结合，用通俗易懂的语言描述系统虚拟化技术原理，其中不乏具有代表性和普遍意义的实例和技术细节，是学习系统虚拟化技术的宝贵资料。本书不仅可以作为教材，供计算机相关专业的大学高年级学生和研究生阅读；而且可以作为一本参考手册，供大学或企业里与系统相关领域的研究开发人员以及对虚拟机及虚拟化核心技术有兴趣的研究者和开源工作者阅读。

图书在版编目(CIP)数据

系统虚拟化：原理与实现/英特尔开源软件技术中心，复旦大学并行处理研究所著. —北京：清华大学出版社，2009.3
ISBN 978-7-302-19372-2

Ⅰ. 系…　Ⅱ. ①英…②复…　Ⅲ. 虚拟技术　Ⅳ. TP391.9

中国版本图书馆CIP数据核字(2009)第011521号

责任编辑：索　梅　徐跃进
责任校对：梁　毅
封面设计：崔爽纯
责任印制：杨　艳
出版发行：清华大学出版社　　**地　　址**：北京清华大学学研大厦A座
http://www.tup.com.cn　　**邮　　编**：100084
社　总　机：010-62770175　　**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 刷 者：北京市清华园胶印厂
装 订 者：三河市溧源装订厂
经　　销：全国新华书店
开　　本：185×230　**印　张**：16　**字　数**：353千字
版　　次：2009年3月第1版　　**印　　次**：2009年3月第1次印刷
印　　数：1～3000
定　　价：29.00元

PREFACE 前言

虚拟化技术在近期成为了学术界和产业界的一大焦点，并且被认为是在将来的一段时间内最具影响力的技术之一，它可能会改变现有系统软件的整个样子，为系统软件带来一场新的革命。

虚拟化技术正在成为系统软件中广泛存在的一层，它的普及可以从三个角度来看待。从硬件平台来讲，虚拟化技术被用于企业级服务器、桌面平台（例如台式计算机和笔记本式计算机）以及嵌入式系统中；从用途来讲，虚拟化技术被用于系统资源管理、容错、软硬件维护、增强系统安全、提升性能和节能等领域；从趋势来讲，虚拟化技术正在广泛地与其他技术结合，并且得到更多硬件上的支持，其性能损失不断降低，部分固化到硬件中。

虚拟化技术的含义很广泛。将任何一种形式的资源抽象成另一种形式的技术都是虚拟化。在常用的操作系统中就存在某种意义上的“虚拟化技术”，例如虚拟内存空间和进程。如果把内存看作是一个设备，虚拟内存就是将物理内存虚拟成多个内存空间。虚拟内存的容量可以少于或多于物理内存。进程的概念实际是对于物理硬件执行环境的一个抽象，每一个进程都享有一个完整的硬件执行环境，并且与其他进程相隔离。

相对于进程级的虚拟化，虚拟机是另外一个层面的虚拟化，即系统级虚拟化。与虚拟单个进程的执行环境所不同，系统级虚拟化所抽象的环境是整个计算机，其抽象出的环境称为虚拟机，包括 CPU、内存和 I/O。在每个虚拟机中都可以运行一个操作系统，在一台计算机上可以虚拟出多个虚拟机。

本书尝试将当前主要的虚拟机和系统级虚拟化原理梳理出来，从一个系统设计者的角度来介绍。从基本的原理出发，本书结合主流的 x86 体系结构和硬件上对虚拟化的支持来介绍系统级虚拟化是如何实现的。除介绍虚拟机与系统级虚拟化原理之外，本书力图加入学术界对于虚拟化技术或利用虚拟化技术的最新研究、产业界的最新应用和将来可能的发展趋向。

1. 面向的读者

系统虚拟化是一门跨领域的学科，涉及操作系统、编译和体系结构等学科知识，并延展到资源管理、性能和系统安全等问题。

本书定位的读者包括计算机相关专业的高年级学生、研究生、研究开发人员及对虚拟机

及虚拟化核心技术有兴趣的学者。

2. 全书结构

本书的结构安排尽可能使每章的内容自包含，尽力让对于某一章节感兴趣的读者不需检索其他章节的内容。

第 1 章从虚拟化技术的历史开始讲起，将现有的虚拟化技术作一个分类。

第 2 章介绍了一个缩略版的计算机系统，帮助读者温习这些知识。其内容主要包括硬件抽象层、操作系统的硬件管理机制以及进程等与后续章节有关的操作系统概念。对于这些内容已经了解的读者可以直接跳过这一章。

第 3 章介绍典型系统级虚拟化的技术以及 VMM 的组成和分类，最后还介绍一些目前市场比较流行的虚拟化产品。

第 4 章介绍基于软件的完全虚拟化技术。

第 5 章介绍硬件辅助的完全虚拟化技术。

第 6 章以 Xen 为例介绍类虚拟化技术的实现原理。

第 7 章介绍虚拟机的性能评测和调试技术。

第 8 章介绍系统虚拟化的应用实例。

第 9 章对虚拟机和系统虚拟化技术的发展作一个展望。

本书的第 1 章由复旦大学张逢喆、英特尔公司董耀祖、李少凡合作撰写；第 2 章由复旦大学俞捷和英特尔公司张鑫合作撰写；第 3 章由英特尔公司田坤和余珂撰写；第 4 章由复旦大学张逢喆和黄弋简撰写；第 5 章由英特尔公司余珂、李欣、蒋运宏和徐雪飞撰写；第 6 章由复旦大学张逢喆、刘鹏程和黄弋简撰写；第 7 章由英特尔公司董耀祖和杨晓伟撰写；第 8 章由英特尔公司余珂、王庆和复旦大学吴曦、袁立威合作完成；第 9 章由复旦大学刘鹏程、周亦勋、宋翔和英特尔公司董耀祖合作完成。英特尔公司李少凡和董耀祖对本书的每一版草稿均作了细致的审阅工作，余珂、张鑫、王庆以及复旦大学的张逢喆对全书的统编和修改作了大量的工作。

3. 如何阅读本书

对于虚拟机和系统虚拟化基本原理可以阅读第 1、3、4、5、6 章。

希望单独了解基于软件的完全虚拟化、硬件辅助的完全虚拟化或类虚拟化的读者可以单独阅读对应的章节。

希望了解系统虚拟化性能评测和优化技术的读者可以阅读第 7 章。

希望了解系统虚拟化技术背景、应用和发展的读者可以阅读第 1、8、9 章。

4. 感谢

在这里，首先要感谢英特尔公司副总裁王文汉博士、英特尔亚太研发有限公司总经理兼首席研发官梁兆柱博士、英特尔亚太研发有限公司创新中心总监黄波博士和英特尔开源技术中心高级经理冯晓焰先生，以及复旦大学软件学院院长臧斌宇教授，他们是本书的发起人，并一直鼓励我们完成本书。

也要谢谢所有在英特尔开源技术中心工作的同事以及所有在复旦大学软件学院学习工

作的同事和同学们，感谢他们不仅在工业界还在学术界推动虚拟化技术向前发展所做的努力，同时也感谢他们对本书草稿进行了一遍又一遍的阅读，并提出了许多定贵的修改意见。在此，特别感谢英特尔公司辛晓慧、崔得暄、韩伟东、贺青和单海涛等，他们为本书提供了大量技术资料。还要特别感谢复旦大学陈海波、陈榕、杨子夜、王慧红和陈诚等，他们为本书的编撰提供了许多帮助。

最后，感谢您在茫茫书海中选择了本书，并衷心祝愿您能从中受益。

虚拟化专题写作组
2008 年 9 月

CONTENTS

目录

CHAPTER 1

第1章 开篇

虚拟化(Virtualization)以各种形式存在已经有四十多年的时间了。它对于不同的人来说可能意味着不同的东西,这要取决于他们所从事工作领域的环境。有经验的程序员可能还记得,曾有一段时间他们担心是否有可用内存来存放自己的程序指令和数据,于是出现了虚拟内存。后来,为了更好地时分共享(Time-sharing)昂贵的大型机系统,出现了虚拟服务器。然而,虚拟化技术的内涵远远不止于虚拟内存和虚拟服务器。目前,已经有了网络虚拟化、微处理器虚拟化、文件虚拟化和存储虚拟化等技术,如果在一个更广泛的环境中或从更高级的抽象(如任务负载虚拟化和信息虚拟化)来思考虚拟化技术,虚拟化技术就变成了一个非常强大的概念,可以为最终用户、应用程序和企业提供很多优点。抽象来说,虚拟化是资源的逻辑表示,它不受物理限制的约束。具体来说,虚拟化技术的实现形式是在系统中加入一个虚拟化层,虚拟化层将下层的资源抽象成另一形式的资源,提供给上层使用。通过空间上的分割、时间上的分时以及模拟,虚拟化可以将一份资源抽象成多份。反过来,虚拟化也可以将多份资源抽象成一份。总的来说,虚拟化可以把一个纷繁复杂、无计划性的世界改造成一个似乎是为人们的特定需求而度身订造的世界。

系统虚拟化是虚拟化技术中的一种,其抽象的粒度是整个计算机。早在20世纪60年代这个名称就已经诞生,从这个程度上来说,这是一个和操作系统有着同样悠久历史的领域。在虚拟化技术发展的几十年历程中,它经历了数次大幅度的起落,人们不断被虚拟化技术潜在的功能所吸引,然后又因客观技术上的限制而放弃。但是,随着近年来处理器技术和性能的迅猛发展,虚拟化技术成熟的时机真正到来。尤其是硬件虚拟化技术的诞生(例如Intel VT和AMD SVM技术),极大地扩展了虚拟化技术的应用范围。本章先简单介绍一些常见的虚拟化概念,然后着重介绍系统虚拟化,包括它的发展历史、特点以及系统虚拟化会带来什么好处。

1.1 形形色色的虚拟化

现代计算机系统是一个庞大的整体，整个系统的复杂性是不言而喻的。因而，计算机系统被分成了多个自下而上的层次。图 1-1 所示的是一种常见的计算机系统中的抽象层。每一个层次都向上一层次呈现一个抽象，并且每一层只需知道下层抽象的接口，而不需要了解其内部运作机制。例如，操作系统所看到的硬件是一个硬件抽象层，而不需要理解硬件的布线或者电气特性等。这样，以层的方式抽象资源的好处是每一层只需要考虑本层设计以及与相邻层间的相互交互，从而大大降低了系统设计的复杂性，提高了软件的移植性。

如图 1-1 所示，在硬件与操作系统之间的是硬件抽象层，在操作系统与应用程序或函数库之间的是 API 抽象层。硬件抽象层（Hardware Abstraction Layer，HAL）是计算机中软件所能控制的硬件的抽象接口，通常包括 CPU 的各种寄存器、内存管理模块、I/O 端口和内存映射的 I/O 地址等。API 抽象层抽象的是一个进程所能控制的系统功能的集合，包括创建新进程、内存申请和归还、进程间同步与共享、文件系统和网络操作等。

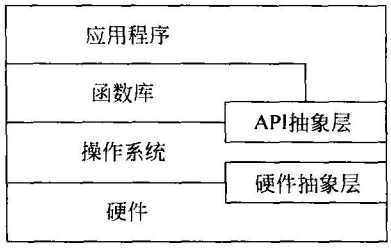


图 1-1 计算机系统的各个抽象层

本质上，虚拟化就是由位于下层的软件模块，通过向上一层软件模块提供一个与它原先所期待的运行环境完全一致的接口的方法，抽象出一个虚拟的软件或硬件接口，使得上层软件可以直接运行在虚拟的环境上。虚拟化可以发生在现代计算机系统的各个层次上，不同层次的虚拟化会带来不同的虚拟化概念。因此，在学术界和工业界里，也先后出现了各种形形色色的虚拟化概念。下面介绍一些常见的虚拟化概念。

在介绍各种虚拟化概念之前，先介绍虚拟化中的两个重要名词。在虚拟化中，物理资源通常有一个定语称为宿主（Host），而虚拟出来的资源通常有一个定语称为客户（Guest）。根据资源的不同，这两个名词的后面可以接不同的名词。例如，如果是将一个物理计算机虚拟为一个或多个虚拟计算机，则这个物理计算机通常也被称为宿主机（Host Machine），而其上运行的虚拟机被称为客户机（Guest Machine）。宿主机上如果运行有操作系统，通常称为宿主机操作系统（Host OS），而虚拟机中运行的操作系统被称为客户机操作系统（Guest OS）。

1. 硬件抽象层上的虚拟化

硬件抽象层上的虚拟化是指通过虚拟硬件抽象层来实现虚拟机，为客户机操作系统呈现和物理硬件相同或相近的硬件抽象层。由于客户机操作系统所能看到的是硬件抽象层，因此，客户机操作系统的行为和在物理平台上没有什么区别。通常来说，宿主机和客户机的

ISA(Instruction Set Architecture,指令集架构)是相同的,客户机的大部分指令可以在宿主处理器上直接运行。只有那些需要虚拟化的指令才会由虚拟化软件进行处理,从而大大降低了虚拟化开销。另外,客户机和宿主机的硬件抽象层的其他部分如中断控制器、设备等,可以是完全不同的,当客户机对硬件抽象层访问时,虚拟化软件需要对此进行截获并模拟。比较知名的硬件抽象层上的虚拟化有 VMware 的系列产品、Xen 等。

2. 操作系统层上的虚拟化

操作系统层上的虚拟化是指操作系统的内核可以提供多个互相隔离的用户态实例。这些用户态实例(经常被称为容器)对于它的用户来说就像是一台真实的计算机,有自己独立的文件系统、网络、系统设置和库函数等。从某种意义上说,这种技术可以被认为是 UNIX 系统 chroot 命令的一种延伸。因为这是操作系统内核主动提供的虚拟化,因此操作系统层上的虚拟化通常非常高效,它的虚拟化资源和性能开销非常小,也不需要有硬件的特殊支持。但它的灵活性相对较小,每个容器中的操作系统通常必须是同一种操作系统。另外,操作系统层上的虚拟化虽然为用户态实例间提供了比较强的隔离性,但其粒度是比较粗的。因为操作系统层上虚拟化的高效性,它被大量应用在虚拟主机服务环境中。比较有名的操作系统级虚拟化解决方案有 Paralles 的 Virtuozzo,Solaris 的 Zone 和 Linux 的 VServer 等。

3. 库函数层上的虚拟化

操作系统通常会通过应用级的库函数提供给应用程序一组服务,例如文件操作服务、时间操作服务等。这些库函数可以隐藏操作系统内部的一些细节,使得应用程序编程更为简单。不同的操作系统库函数有着不同的服务接口,例如 Linux 的服务接口是不同于 Windows 的。库函数层上的虚拟化就是通过虚拟化操作系统的应用级库函数的服务接口,使得应用程序不需要修改,就可以在不同的操作系统中无缝运行,从而提高系统间的互操作性。例如,WINE 系统是在 Linux 上模拟了 Windows 的库函数接口,使得一个 Windows 的应用程序能够在 Linux 上正常运行。

4. 编程语言层上的虚拟化

另一大类编程语言层上的虚拟机称为语言级虚拟机,例如 JVM (Java Virtual Machine)和微软的 CLR(Common Language Runtime)。这一类虚拟机运行的是进程级的作业,所不同的是这些程序所针对的不是一个硬件上存在的体系结构,而是一个虚拟体系结构。这些程序的代码由虚拟机的运行时支持系统首先翻译为硬件的机器语言,然后再执行。通常一个语言类虚拟机是作为一个进程在物理计算机系统中运行的,因此,它属于进程级虚拟化。

1.2 系统虚拟化

系统虚拟化是指将一台物理计算机系统虚拟化为一台或多台虚拟计算机系统。每个虚拟计算机系统(简称为虚拟机)都拥有自己的虚拟硬件(如 CPU、内存和设备等),来提供一

个独立的虚拟机执行环境。通过虚拟化层的模拟，虚拟机中的操作系统认为自己仍然是独占一个系统在运行。每个虚拟机中的操作系统可以完全不同，并且它们的执行环境是完全独立的。这个虚拟化层被称为虚拟机监控器(Virtual Machine Monitor，VMM)。如图 1-2 所示。

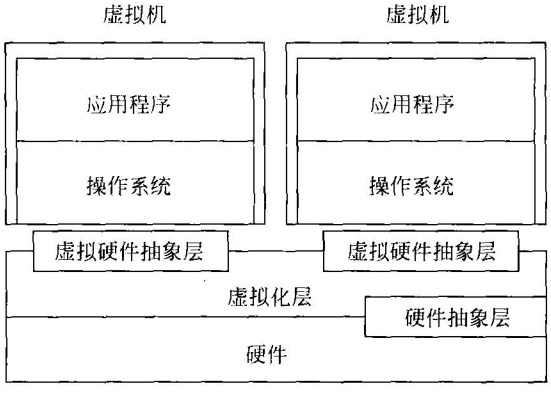


图 1-2　系统虚拟化

从本质上来说，虚拟计算机系统和物理计算机系统可以是两个完全不同 ISA 的系统。例如，可以在一个 x86 的物理计算机上运行一个安腾的虚拟计算机。但是，不同的 ISA 使得虚拟机的每一条指令都需要在物理机上模拟执行，从而造成性能上的极大下降。当然，相同体系结构的系统虚拟化通常会有比较好的性能，VMM 实现起来也会比较简单。虚拟机的大部分指令可以在处理器上直接运行，只有那些需要虚拟化的指令才会由 VMM 进行处理。

1974 年，Popek 和 Goldberg 定义了虚拟机可以看作是物理机的一种高效隔离的复制。上面的定义里蕴涵了三层含义(同质、高效和资源受控)，这也是一个虚拟机所具有的三个典型特征。

所谓同质是指虚拟机的运行环境和物理机的环境在本质上需要是相同的，但是在表现上能够有一些差异。例如，虚拟机所看到的处理器个数可以和物理机上实际的处理器个数不同，处理器主频也可以与物理机的不同，但是物理机上和虚拟机中看到的处理器必须是同一种基本类型的。

所谓高效是指虚拟机中运行的软件需要有接近在物理机上直接运行的性能。为了做到这一点，软件在虚拟机中运行时，大多数的指令需要直接在硬件上执行，只有少量指令需要经过 VMM 处理或模拟。

所谓资源受控是指 VMM 需要对系统资源有完全控制能力和管理权限，包括资源的分配、监控和回收。

系统虚拟化最早出现于 20 世纪 60 年代的 IBM System360 上。经过一段时间的沉寂