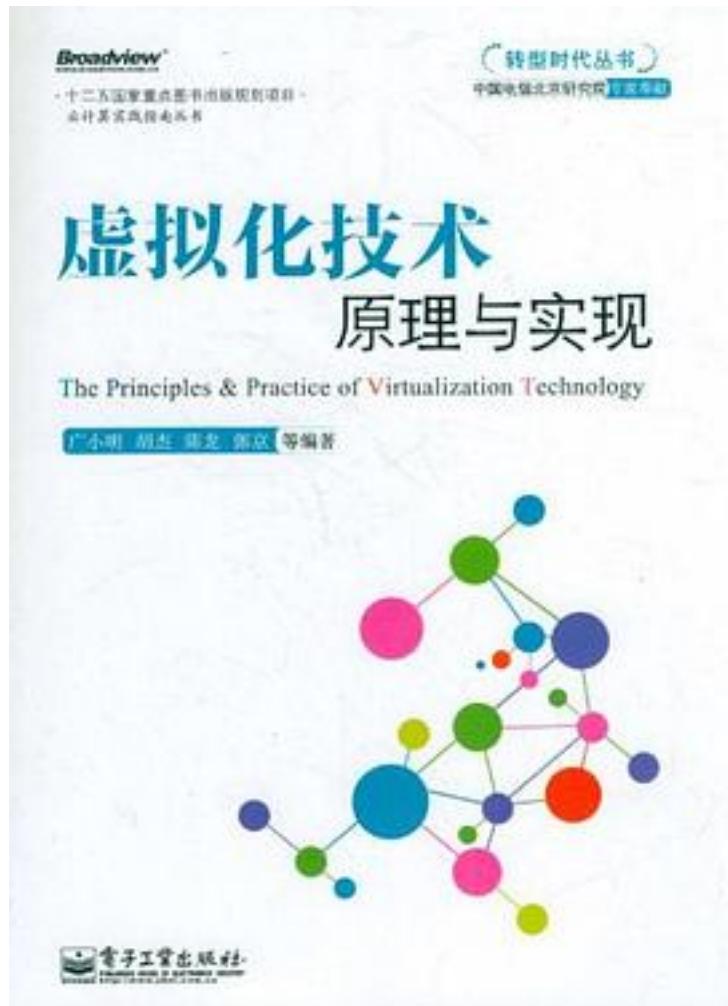


虚拟化技术原理与实现



[虚拟化技术原理与实现_下载链接1](#)

著者:广小明

出版者:

出版时间:2012-10

装帧:

isbn:9787121185281

《虚拟化技术原理与实现》对云计算中关键技术之一的虚拟化技术进行了深入的分析，

从x86计算机体系结构以及操作系统的工作原理出发，介绍了虚拟化技术原理以及业界主流虚拟化软件产品，并以Xen、KVM开源软件为例分析了虚拟化软件的架构及其实现方法，最后对虚拟化软件管理接口的工作原理以及实现方法进行了全面的梳理。

作者介绍：

目录: 第一篇 云计算与虚拟化技术 1

第1章 虚拟化技术基本原理 2

1.1 云计算与虚拟化技术 3

1.2 x86和非x86体系结构基础 4

1.2.1 x86的发展历程 4

1.2.2 x86-64 6

1.2.3 x86内存架构 7

1.2.4 x86-64的基本模式 23

1.2.5 x86-64的寄存器组 25

1.2.6 中断与异常 26

1.2.7 I/O架构 31

1.2.8 DMA 32

1.2.9 时钟 33

1.3 操作系统与虚拟化 34

1.3.1 操作系统 35

1.3.2 进程 35

1.3.3 系统虚拟化 38

1.3.4 系统虚拟化的发展历程 39

1.3.5 可虚拟化条件 41

1.3.6 虚拟化的原理与分类 43

1.4 VMM技术架构分类 45

1.4.1 Hypervisor模型 46

1.4.2 宿主(Hosted)模型 47

1.4.3 混合模型 48

1.5 本章小结 49

第2章 虚拟化实现技术架构 50

2.1 处理器虚拟化实现技术 52

2.1.1 Intel VT 53

2.1.2 AMD SVM 55

2.1.3 vCPU 55

2.2 中断虚拟化实现技术 56

2.3 内存虚拟化实现技术 58

2.3.1 影子页表 60

2.3.2 Intel EPT 65

2.3.3 AMD NPT 67

2.4 I/O设备虚拟化实现技术 68

2.4.1 Intel VT-d 69

2.4.2 DMA重映射 70

2.4.3 I/O页表 73

2.4.4 AMD IOMMU 74

2.5 网络虚拟化技术 76

2.6 时间虚拟化技术 79

2.6.1 操作系统和客户机的时间概念 79

2.6.2 客户机时间概念的实现 82

2.7 主流虚拟化产品及其特点 84

2.7.1 Xen 84

2.7.2 VMware	86
2.7.3 Hyper-V	87
2.7.4 KVM	88
2.8 本章小结	90
第二篇 Xen虚拟化技术	91
第3章 Xen软件系统原理	92
3.1 Xen软件模块结构	93
3.1.1 Xen Hypervisor	93
3.1.2 特权虚拟域0 (Dom0)	94
3.1.3 独立设备驱动域 (IDD)	95
3.1.4 非特权虚拟域U (DomU)	96
3.1.5 硬件虚拟域 (HVM)	96
3.2 Xen系统启动工作原理及流程	96
3.2.1 系统引导过程	97
3.2.2 Hypervisor启动与初始化过程	98
3.2.3 Dom0启动过程	99
3.2.4 DomU的启动	99
3.3 Xen CPU虚拟化工作原理	100
3.3.1 x86体系虚拟化存在的问题	100
3.3.2 CPU虚拟化——半虚拟化 (又称为泛虚拟化)	102
3.3.3 CPU虚拟化技术——硬件虚拟化技术支持的全虚拟化	103
3.4 Xen内存虚拟化工作原理	105
3.4.1 内存虚拟化——直接模式	106
3.4.2 内存虚拟化——影子模式	107
3.5 I/O虚拟化工作原理	108
3.5.1 半虚拟化I/O	108
3.5.2 全虚拟化I/O	109
3.6 Xen虚拟机 (DomU) 生命周期管理	110
3.7 本章小结	112
第4章 Xen Hypervisor技术实现	113
4.1 Xen Hypervisor关键技术概述	114
4.2 Hypercall	114
4.2.1 Hypercall的实现机制	115
4.2.2 自定义Hypercall的方法	118
4.2.3 应用程序使用Hypercall的方法	120
4.3 事件通道	121
4.3.1 事件通道的初始化	121
4.3.2 事件通道的绑定	122
4.3.3 发送事件通知	136
4.3.4 事件通知的处理	138
4.4 数据共享	142
4.4.1 授权表 (Grant table)	142
4.4.2 XenStore和XenBus	146
4.4.3 分离设备驱动	149
4.5 本章小结	154
第三篇 KVM虚拟化技术	155
第5章 qemu-kvm虚拟化解决方案	156
5.1 概述	157
5.2 内核模块组成概述	158
5.2.1 KVM的内核模块结构	158
5.2.2 Linux内核源码中的KVM	160
5.3 KVM所提供的API	162
5.3.1 KVM API纵览	162
5.3.2 system ioctls调用	163

5.3.3 vm ioctl系统调用	164
5.3.4 vcpu ioctl系统调用	165
5.4 KVM内核模块重要的数据结构	168
5.4.1 KVM结构体	168
5.4.2 kvm_vcpu结构体	169
5.4.3 kvm_x86_ops结构体	169
5.4.4 KVM API中重要的结构体	171
5.5 KVM内核模块重要流程的分析	173
5.5.1 初始化流程	173
5.5.2 虚拟机的创建	175
5.5.3 vCPU的创建	177
5.5.4 vCPU的运行	180
5.6 qemu-kvm软件架构分析	184
5.6.1 QEMU的三种运行模式	184
5.6.2 libvirt和virt-manager	185
5.6.3 KVM的调试接口	186
5.7 本章小结	187
第6章 qemu-kvm原理与分析	188
6.1 QEMU软件架构	189
6.1.1 qemu-kvm的配置与编译	189
6.1.2 qemu-kvm的架构与配置	190
6.2 QEMU组件	190
6.2.1 模块模型	190
6.2.2 libkvm模块	193
6.2.3 virtio组件	196
6.3 基于KVM的QEMU PC Emulator	199
6.3.1 KVM中的Machine模块	199
6.3.2 基于KVM加速支持的CPU虚拟化模块	207
6.3.3 虚拟机的内存管理	216
6.3.4 I/O管理	223
6.4 本章小结	225
第四篇 虚拟化软件开放接口	227
第7章 Xen API接口技术及实现	228
7.1 Xen Management API接口概述	229
7.2 XML-RPC工作原理	230
7.2.1 XML-RPC概述	231
7.2.2 XML-RPC请求	232
7.2.3 XML-RPC响应	234
7.3 Xen Managemnet API的实现	236
7.3.1 C语言和Python语言的扩展与嵌入	236
7.3.2 Xen Management API类的定义	237
7.3.3 Xen Management API处理流程分析	238
7.4 本章小结	242
第8章 libvirt虚拟化控制中间件	243
8.1 libvirt概述	244
8.1.1 libvirt简介及使用样例	244
8.1.2 基于libvirt所开发的开源应用	245
8.1.3 安装与配置	245
8.2 libvirt架构与开发	247
8.2.1 libvirt架构说明	247
8.2.2 libvirt API控制接口	250
8.2.3 libvirt的主机域管理	254
8.2.4 libvirt的网络架构	254
8.2.5 libvirt的存储管理	256

8.3 基于libvirt的XML配置解析	256
8.3.1 XML配置格式简析	256
8.3.2 针对Xen的libvirt配置详解	264
8.3.3 针对KVM/QEMU的libvirt配置详解	271
8.4 本章小结	282
参考文献	283
• • • • • (收起)	

[虚拟化技术原理与实现](#) [下载链接1](#)

标签

虚拟化

操作系统

虚拟化技术原理与实现

云计算

计算机科学

计算机

程序设计

libvirt

评论

1/3的内容是抄的，还是整段整章的抄，参考文献一；1/3直接是代码；而且还这么贵，50块钱，300页的小册子。

对普及虚拟化知识还是值得一读的，至少知道一些概念。

和容器技术的使用场景有重叠，虚拟化相比容器(分割资源)是更底层的技术(模拟硬件)，但两者的界限会渐渐模糊，因为软硬件的功能界限都是越来越模糊了

大致读完了，对于了解一些基础原理还是有用的。

基础不够全面，深度又不够。

这是一本很好的入门书

虚拟化原理部分很多句子很眼熟。前段时间才看过实验室写的书：《虚拟化原理》，所以知道这本书“摘抄”了很多那本书的段落。还好在reference部分把《虚拟化原理》放在第一位的。详细讲解kvm的部分除了代码，其他内容是IBM技术博客上的入门内容。内容简介上说的“对虚拟化管理接口工作原理及实现方法进行了全面的梳理”我觉得言过其实了。

写的相当的虚，全是肤浅的理论，感觉作者也并没有完全理解。

[虚拟化技术原理与实现 下载链接1](#)

书评

[虚拟化技术原理与实现 下载链接1](#)