

Xen虚拟化技术完全导读



[Xen虚拟化技术完全导读_下载链接1](#)

著者:大卫 (Chisnall David)

出版者:北京航空航天大学出版社

出版时间:2014-1-1

装帧:平装

isbn:9787811245639

本书主要介绍了目前IT技术热点虚拟化技术领域中最受关注的虚拟化系统软件Xen，包

括在Xen中对于各种虚拟化技术的实现的分析，尤其基于研究热点的考虑用大量篇幅专门讨论了虚拟化的I/O和Xen内核的一些关键技术，并讨论了未来的方向，其中又以特别的章节分析了硬件虚拟化（HVM）。

读者对象以研究虚拟化技术的科研人员和工程人员为主，尤其是从事系统软件分析和开发的以及服务器端高可靠性软件研发的人员。

Chisnall

David编著的《Xen虚拟化技术完全导读》是一本全面讲述XenSource功能强大的准虚拟化解方案内部机制的完全手册。从架构到内核，作者David Chisnall罗列关键构件代码，向读者展示这种技术方案如何工作，为读者提供需要充分利用以及应用Xen管理程序开发低成本、高性能的Linux和Windows虚拟环境的重要资讯。

为准许独立访问XenSource团队，Chisnall针对虚拟化技术概述和Xen管理程序背后的设计理念制定了一个坚实的框架。紧接着，Chisnall带领读者深入探索管理程序的架构、界面、设备支持、管理工具以及开发人员对虚拟环境优化应用的关键信息所涵盖的内部构件。他以真实示例和现成的运用来揭示Xen的强大和隐患。以此读者可以通过学习获得宝贵的经验。

这些具有深刻见解的资源向读者展示出一幅Xen所有的管理程序究竟是怎样来适应并一起工作的详细场景，让读者置身于如何构建和实施一个精简的、成本效益的虚拟企业。Xen虚拟架构的认知。

作者介绍:

作者:(美)Chisnall David 译者:张炯、吕紫旭、胡彦彦、文成建

目录: 第1章 虚拟化技术的现状1

1.1 什么是虚拟化技术1

1.1.1 CPU的虚拟化2

1.1.2 I/O的虚拟化3

1.2 为什么要虚拟化4

1.3 历史上第一台虚拟机5

1.4 x86架构虚拟化的问题6

1.5 一些解决x86架构虚拟化问题的方案6

1.5.1 二进制翻译7

1.5.2 泛虚拟化8

1.5.3 硬件辅助虚拟化10

1.6 Xen的理念11

1.6.1 方案和机制的分离11

1.6.2 做得越少越好12

1.7 Xen的系统结构13

1.7.1 Hypervisor，操作系统，应用程序之间的关系13

1.7.2 Domain 0的角色15

1.7.3 非特权级的Domain17

1.7.4 HVM的Domain18

1.7.5 Xen的结构配置18

第2章 探索Xen虚拟体系结构22

2.1 作为泛虚拟化客户端启动22

2.2 利用特权级限制操作23

2.3 用超级调用取代特权指令24

2.4 探索Xen事件模型	27
2.5 与共享内存进行通信	28
2.6 拆分设备驱动模型	29
2.7 VM生命周期	30
2.8 练习：最简单的Xen内核	31
2.8.1 客户机入口点	33
2.8.2 把所有内容放在一起	35
第3章 理解Shared Info Pages	39
3.1 获取启动时钟信息	39
3.2 Shared Info Page	42
3.3 Xen中的时间管理	44
3.4 练习：实现函数gettimeofday()	45
第4章 使用授权表 (Grant Table)	49
4.1 内存共享	49
4.1.1 映射 (Mapping) 一个页面	51
4.1.2 domain间的数据传递 (Transferring)	52
4.2 设备I/O环	54
4.3 授权以及撤销授权	55
4.4 练习：映射授权页面(granted page)	57
4.5 练习：在VM之间共享内存	59
第5章 Xen的内存管理	62
5.1 x86环境下的内存管理	62
5.2 伪物理地址模型 (Pseudo-Physical Memory Model)	65
5.3 32位x86系统中的分段模式	66
5.4 使用Xen Memory Assist	67
5.5 使用Ballon Driver控制内存使用	69
5.6 其他内存操作	70
5.7 更新页表	73
5.7.1 创建新的虚拟机(VM)实例	77
5.7.2 处理页故障	77
5.7.3 暂停 (suspend)、恢复和迁移	78
5.8 练习：映射Shared Info Page	79
第6章 理解设备驱动	80
6.1 分离设备模型	80
6.2 将驱动程序移出Domain	82
6.3 理解共享存储器环形缓冲区	83
6.3.1 分析Xen的实现	85
6.3.2 通过内存栅障 (Memory Barriers) 实现顺序操作	87
6.4 通过XenBus连接设备	88
6.5 处理来自消息的通知	90
6.6 通过XenStore进行配置	90
6.7 练习：控制台设备	91
第7章 使用事件通道	96
7.1 事件和中断	96
7.2 处理陷阱 (Trap)	96
7.3 事件类型	99
7.4 请求事件	100
7.5 绑定事件通道到VCPU上	103
7.6 绑定通道上的操作	104
7.7 获取通道状态	105
7.8 屏蔽事件	106
7.9 事件和调度	108
7.10 示例：一个完整的控制台驱动	109
第8章 深入学习XenStore	116

- 8.1 XenStore接口116
- 8.2 浏览XenStore117
- 8.3 XenStore设备119
- 8.4 读和写一个键121
 - 8.4.1 用户空间方法122
 - 8.4.2 从内核调用的方法124
- 8.5 其他操作132
- 第9章 支持核心设备133
 - 9.1 虚拟块设备驱动133
 - 9.1.1 设置块设备134
 - 9.1.2 数据传输136
 - 9.2 使用Xen网络139
 - 9.2.1 虚拟网络接口驱动140
 - 9.2.2 设置虚拟接口140
 - 9.2.3 发送和接收141
 - 9.2.4 NetChannel2144
- 第10章 其他Xen设备147
 - 10.1 CD的支持147
 - 10.2 虚拟帧缓冲器 (Frame Buffer) 147
 - 10.3 TPM驱动152
 - 10.4 原生硬件152
 - 10.4.1 PCI支持153
 - 10.4.2 USB设备155
 - 10.5 添加新的设备类型155
 - 10.5.1 广播设备156
 - 10.5.2 设置环形缓冲区156
 - 10.5.3 困难157
 - 10.5.4 访问设备158
 - 10.5.5 设计后端驱动159
- 第11章 Xen API162
 - 11.1 XML-RPC162
 - 11.1.1 XML-RPC数据类型162
 - 11.1.2 远程过程调用164
 - 11.2 探索Xen接口层次164
 - 11.3 Xen API类166
 - 11.4 Xend的功能169
 - 11.5 Xm命令行171
 - 11.6 Xen CIM提供者172
 - 11.7 练习：枚举正在运行的虚拟机173
 - 11.8 总结177
- 第12章 虚拟机调度178
 - 12.1 调度器接口概述178
 - 12.2 历史上的调度器180
 - 12.2.1 SEDF181
 - 12.2.2 Credit调度器182
 - 12.3 使用调度器API183
 - 12.3.1 运行一个调度器184
 - 12.3.2 Domain 0交互187
 - 12.4 练习：添加一个新的调度器188
 - 12.5 总结191
- 第13章 HVM192
 - 13.1 运行未经修改的操作系统192
 - 13.2 Intel VT-x和AMD SVM194
 - 13.3 HVM设备支持195

13.4 混合虚拟化	196
13.5 BIOS仿真	199
13.6 设备模型和传统的I/O仿真	200
13.7 半虚拟化I/O	201
13.8 Xen中HVM支持	202
第14章 未来的发展方向	206
14.1 真实到虚拟，周而复始	206
14.2 仿真和虚拟化	207
14.3 移植的努力	207
14.4 桌面	209
14.5 功耗管理	211
14.6 关于Domain0的问题	213
14.7 Stub域	215
14.8 新的设备	216
14.9 特殊的架构	217
14.10 前景	219
附录 泛虚拟化客户操作系统移植概述	221
A.1 Domain创建工具（Domain Builder）	221
A.2 启动环境	222
A.3 设置虚拟中断描述符表（IDT）	222
A.4 页表管理	223
A.5 驱动	223
A.6 Domain0的责任	224
A.7 效率	224
A.8 小结	225
• • • • •	(收起)

[Xen虚拟化技术完全导读_下载链接1](#)

标签

虚拟化

Xen

计算机

虚拟机

前沿技术

xen

评论

还是翻译太差

翻过，无意向

[Xen虚拟化技术完全导读_下载链接1](#)

书评

原版应该还不错。就是翻译质量一般。语句一看就是机器翻译的，前后术语不一致，就是用来几小时快速翻翻。
好处是篇幅不大，简单明了，还有例子程序，适合程序员快速了解XEN的基本组成。但是翻译太差，没有办法准确理解各个术语、基本概念，不值得多看。

[Xen虚拟化技术完全导读_下载链接1](#)