

# Oracle® Solaris 10 System Virtualization Essentials

ORACLE®  
SOLARIS



Jeff Victor • Jeff Savit • Gary Combs • Simon Hayler • Bob Netherton

Many of the designations used by manufacturers and sellers to distinguish their products are claimed as trademarks. Where those designations appear in this book, and the publisher was aware of a trademark claim, the designations have been printed with initial capital letters or in all capitals.

Oracle and Java are registered trademarks of Oracle and/or its affiliates. Other names may be trademarks of their respective owners.

AMD, Opteron, the AMD logo, and the AMD Opteron logo are trademarks or registered trademarks of Advanced Micro Devices. Intel and Intel Xeon are trademarks or registered trademarks of Intel Corporation. All SPARC trademarks are used under license and are trademarks or registered trademarks of SPARC International, Inc. UNIX is a registered trademark licensed through X/Open Company, Ltd.

The authors and publisher have taken care in the preparation of this book, but make no expressed or implied warranty of any kind and assume no responsibility for errors or omissions. No liability is assumed for incidental or consequential damages in connection with or arising out of the use of the information or programs contained herein.

This document is provided for information purposes only and the contents hereof are subject to change without notice. This document is not warranted to be error-free, nor subject to any other warranties or conditions, whether expressed orally or implied in law, including implied warranties and conditions of merchantability or fitness for a particular purpose. We specifically disclaim any liability with respect to this document and no contractual obligations are formed either directly or indirectly by this document. This document may not be reproduced or transmitted in any form or by any means, electronic or mechanical, for any purpose, without our prior written permission.

The publisher offers excellent discounts on this book when ordered in quantity for bulk purchases or special sales, which may include electronic versions and/or custom covers and content particular to your business, training goals, marketing focus, and branding interests. For more information, please contact:

U.S. Corporate and Government Sales  
(800) 382-3419  
corpsales@pearsontechgroup.com

For sales outside the United States, please contact:

International Sales  
international@pearsoned.com

Visit us on the Web: [informit.com/ph](http://informit.com/ph)

*Library of Congress Cataloging-in-Publication Data*

Oracle Solaris 10 system virtualization essentials / Jeff Victor ... [et al.].

p. cm.

Includes index.

ISBN 0-13-708188-X (pbk. : alk. paper)

1. Virtual computer systems. 2. Solaris (Computer file) I. Victor, Jeff.

QA76.9.V5O73 2010

005.4'3--dc22

2010025500

Copyright © 2011, Oracle and/or its affiliates. All rights reserved.

500 Oracle Parkway, Redwood Shores, CA, 94065

Printed in the United States of America. This publication is protected by copyright, and permission must be obtained from the publisher prior to any prohibited reproduction, storage in a retrieval system, or transmission in any form or by any means, electronic, mechanical, photocopying, recording, or likewise. For information regarding permissions, write to:

Pearson Education, Inc.  
Rights and Contracts Department  
501 Boylston Street, Suite 900  
Boston, MA 02116  
Fax: (617) 671-3447

ISBN-13: 978-0-13-708188-2

ISBN-10: 0-13-708188-X

Text printed in the United States on recycled paper at RR Donnelley in Crawfordsville, Indiana.

First printing, August 2010



# Contents

---

Foreword	xi
Preface	xv
Acknowledgments	xxi
About the Authors	xxiii
<b>Chapter 1</b> Introduction to Virtualization	1
1.1 Definitions and Motivations	1
1.1.1 What Is Virtualization?	1
1.1.2 Why Virtualize?	2
1.1.3 Why Is Virtualization so Important for Consolidating Workloads?	5
1.1.4 Other Reasons for Virtualization	7
1.1.5 Common Concepts	8
1.1.6 Use Cases for Virtualization	9
1.2 System Virtualization Models	26
1.2.1 Hardware Partitioning	28
1.2.2 Virtual Machines	32
1.2.3 Operating System Virtualization	36
1.3 Summary	41
<b>Chapter 2</b> Hard Partitioning: Dynamic Domains	43
2.1 Partitions	44
2.1.1 Hardware Partitions	44
2.1.2 The M-Series	44

2.2	Domain Implementation	44
2.2.1	Domain Configurations	48
2.2.2	Domain Combinations	54
2.2.3	Domain Availability	57
2.3	Managing Domains	59
2.3.1	Building Domains Examples	60
2.3.2	View from the Domain	69
2.3.3	Fault Isolation	73
2.3.4	Dynamic Reconfiguration	73
2.3.5	Extending Dynamic Domains	75
2.4	Summary	76
<b>Chapter 3</b>	<b>Oracle VM Server for SPARC</b>	<b>77</b>
3.1	Overview of Logical Domains Features	77
3.1.1	Isolation	78
3.1.2	Compatibility	78
3.1.3	Real and Virtual CPUs	78
3.2	Logical Domains Implementation	80
3.2.1	Domain Roles	80
3.2.2	Dynamic Reconfiguration	82
3.2.3	Virtual I/O	82
3.3	Details of Domain Resources	84
3.3.1	Virtual CPUs	84
3.3.2	Virtual Network Devices	85
3.3.3	Virtual Disk	86
3.3.4	Console and OpenBoot	88
3.3.5	Cryptographic Accelerator	88
3.3.6	Memory	89
3.3.7	Binding Resources to Domains	89
3.4	Installing Logical Domains and Building a Guest Domain	89
3.4.1	Verifying and Installing Firmware	90
3.4.2	Installing Logical Domains Software	90
3.4.3	Configuring the Control Domain	92
3.4.4	Network Connectivity Between Primary and Guest Domains	93
3.4.5	Creating a Domain and Installing Oracle Solaris	94
3.4.6	Viewing a Domain	96
3.4.7	Installing Oracle Solaris into a Domain	97
3.4.8	Observing Guest Domains from the Control Domain	98
3.4.9	Viewing a Domain from the Inside	100
3.4.10	Dynamic Reconfiguration	101
3.4.11	Dynamic Resource Management	102
3.4.12	Cloning a Domain	103

3.5	Domain Mobility	104
3.6	Physical to Virtual Conversion	106
3.7	Ease-of-Use Enhancements	108
3.8	Comparison with Oracle Solaris Containers	110
3.9	Summary	110
<b>Chapter 4</b>	<b>Oracle Solaris 10 as an x86 Guest</b>	<b>113</b>
4.1	Overview	114
4.1.1	Overview of the Hardware Compatibility List	114
4.1.2	Type 1 Hypervisor Overview	114
4.1.3	Xen Open-Source Project-Based Hypervisors	115
4.2	Oracle VM Server for x86	116
4.2.1	Oracle VM Key Features	117
4.2.2	Oracle Solaris PV Drivers	117
4.2.3	Oracle VM Support for Oracle Solaris	118
4.3	xVM Hypervisor	118
4.3.1	xVM Hypervisor Key Features	118
4.3.2	Oracle Solaris PV Drivers	119
4.3.3	xVM Hypervisor Support for Oracle Solaris	119
4.4	Citrix XenServer	120
4.4.1	Citrix XenServer Key Features	120
4.4.2	Oracle Solaris PV Drivers	120
4.4.3	Citrix XenServer Support for Oracle Solaris	121
4.5	VMware ESX	121
4.5.1	VMware vSphere Key Features	122
4.5.2	Oracle Solaris PV Drivers	122
4.5.3	ESX Support for Oracle Solaris	123
4.6	Microsoft Windows Server 2008 Hyper-V	123
4.6.1	Microsoft Windows Server 2008 R2 Hyper-V Key Features	124
4.6.2	Oracle Solaris PV Drivers	124
4.6.3	Microsoft Windows Server 2008 Hyper-V R2 Support for Oracle Solaris	125
4.7	Red Hat Enterprise Virtualization	125
4.7.1	Red Hat Enterprise Virtualization Key Features	126
4.7.2	Oracle Solaris PV Drivers	127
4.7.3	Red Hat Enterprise Virtualization Support for Oracle Solaris	127
4.8	Summary	127
<b>Chapter 5</b>	<b>Oracle VM VirtualBox</b>	<b>129</b>
5.1	How Oracle VM VirtualBox Works	131

5.1.1	Oracle VM VirtualBox Architecture	132
5.1.2	Interacting with Oracle VM VirtualBox	133
5.2	Oracle VM VirtualBox Guest Platform	134
5.2.1	Virtual CPUs	134
5.2.2	RAM	136
5.2.3	Virtual Disk	137
5.2.4	Virtual Network Devices	140
5.2.5	BIOS and EFI	143
5.2.6	Guest Additions	144
5.3	Oracle Solaris as an Oracle VM VirtualBox Host	145
5.3.1	Installing Oracle VM VirtualBox	147
5.4	Oracle Solaris as an Oracle VM VirtualBox Guest	149
5.5	Creating and Managing Oracle VM VirtualBox Guests	151
5.5.1	Creating the Guest Machine	151
5.5.2	Installing the Guest Operating System	156
5.5.3	Creating a Guest Machine Using the Command Line	158
5.5.4	Starting a Virtual Machine	161
5.5.5	Stopping a Virtual Machine	162
5.5.6	Cloning a Virtual Machine	163
5.5.7	Live Migration of a Guest	166
5.6	Summary	167
<b>Chapter 6</b>	<b>Oracle Solaris Containers</b>	<b>169</b>
6.1	Feature Overview	170
6.1.1	Basic Model	171
6.1.2	Isolation	173
6.1.3	Namespaces	177
6.1.4	Brands	177
6.1.5	Packaging, Deployment, and File Systems	178
6.1.6	Patching	180
6.2	Feature Details	181
6.2.1	Container Creation	181
6.2.2	Resource Management	189
6.2.3	Networking	213
6.2.4	Direct Device Access	215
6.2.5	Virtualization Management Features	216
6.3	Solaris 8 Containers and Solaris 9 Containers	219
6.3.1	Oracle Solaris Trusted Extensions	221
6.4	Network Virtualization in OpenSolaris	222
6.5	Strengths of Oracle Solaris Containers	225
6.6	Summary	226

<b>Chapter 7</b>	<b>Choosing a Virtualization Technology</b>	<b>227</b>
7.1	Review of Strengths and Limitations	227
7.1.1	Hard Partitioning and Dynamic Domains	230
7.1.2	Oracle VM Server for SPARC	231
7.1.3	x86 Hypervisors	232
7.1.4	Oracle Solaris Containers	233
7.2	Choosing the Technology: Start with Requirements	235
7.3	Virtualization Decision Tree	236
7.3.1	SPARC Environments	236
7.3.2	x86 Environments	237
7.4	Choosing the Technology: Examples	237
7.4.1	Consolidating Large Workloads	237
7.4.2	Hosting an ISP Web Service	238
7.4.3	Diverse OS Consolidation	239
7.5	Summary	239
<b>Chapter 8</b>	<b>Applying Virtualization</b>	<b>241</b>
8.1	How to Configure for Dynamic Domains	241
8.1.1	M9000 Configuration Example	242
8.1.2	Summary	254
8.2	Consolidating with Oracle VM Server for SPARC (Logical Domains)	254
8.2.1	Planning	255
8.2.2	Configuring Logical Domains	256
8.2.3	Creating Domains	256
8.2.4	Testing	257
8.2.5	Summary	257
8.3	Deploying Oracle Solaris 10 with Oracle VM Server for x86	258
8.3.1	Prerequisites for Installing Oracle Solaris 10 in an Oracle VM Environment	258
8.3.2	Creating an Oracle Solaris 10 Guest Virtual Machine	259
8.3.3	Summary	268
8.4	How to Enable xVM Hypervisor Live Migration	268
8.4.1	Technical Briefs	268
8.4.2	Live Migration Prerequisites	269
8.4.3	Step 1: Configuring the xVM Hypervisor	269
8.4.4	Step 2: Configure Access to a Shared NFS Resource	272
8.4.5	Step 3: Create an Oracle Solaris 10 HVM+PVIO Guest	273
8.4.6	Step 4: Configure and Enable Live Migration	274
8.4.7	Summary	276
8.5	Running Microsoft Windows in an Oracle Solaris Container	276

8.5.1	Planning	277
8.5.2	Configuring the Oracle Solaris Global Zone	278
8.5.3	Creating the Container	278
8.5.4	Creating the Oracle VM VirtualBox Guest Machine User	280
8.5.5	Configuring the Windows Guest Machine	281
8.5.6	Creating an Autostart Service for the Guest	282
8.5.7	Cloning the Windows Container	283
8.5.8	Summary	285
8.6	Consolidating with Oracle Solaris Containers	285
8.6.1	Planning	286
8.6.2	Configure Oracle Solaris 10	287
8.6.3	Create Containers	287
8.6.4	Testing	290
8.6.5	Summary	290
8.7	Security Hardening with Oracle Solaris Containers	290
8.7.1	Scenario	292
8.7.2	Basic Steps	293
8.7.3	Implementing Hardened Containers	294
8.7.4	Security Analysis	298
8.7.5	Summary	299
8.7.6	Further Reading	299
8.8	Summary	300
<b>Chapter 9</b>	<b>Virtualization Management</b>	<b>301</b>
9.1	VE Life-Cycle Management	303
9.1.1	Life Cycle of a VE	303
9.2	Opportunities for Business Agility and Operational Flexibility	306
9.2.1	Problems	307
9.2.2	Virtualization Offers New Opportunities	309
9.3	Oracle Enterprise Manager Ops Center	315
9.3.1	Basic Structure	315
9.3.2	Concepts	317
9.3.3	Secure or Isolated Data Centers	318
9.3.4	Discovering Assets	318
9.3.5	Provisioning Assets	320
9.3.6	Updating Assets	322
9.3.7	Monitoring Assets	323
9.3.8	Managing Assets	325
9.3.9	Oracle EM Ops Center Summary	326
9.4	Summary	326
Appendix: History of Virtualization and Architectural Evolution		327
Index		345





# Foreword

---

I'm no longer sure when I first became hooked. Was it when I overheard a casual conversation about running a “test” copy of MVS in parallel with the real copy of MVS on a new 390 mainframe? Or by the idea of Zarniwoop researching the *Hitchhiker's Guide to the Galaxy* in an electronically synthesized copy of the entire universe he kept in his office? Whatever the cause, I'm still addicted to virtual machine technology.

Fooing a whole stack of software to run correctly on a software simulation of the platform it was designed to run on has been a recurring interest in my career. Poring through the history of VM/370 as an graduate student, absorbing James Gosling's audacious idea of the Java VM, spending a few weeks building an experimental machine emulator to run SPARC applications on Solaris for PowerPC, the “aha!” moment when we realized how useful it would be if we arranged that a set of processes could behave as a little OS within an OS (the idea that became Solaris Zones), the first bring-up of OpenSolaris running as a paravirtualized guest on Xen—those are just a few of the highlights for me.

This book began as a project within Sun in mid-2009 during Oracle's acquisition of the company, so it both explores aspects of Sun's virtualization technology portfolio, and—now that the acquisition is complete—peers a little into 2010. Sun's unique position as a systems company allowed it to deliver a full set of integrated virtualization technologies. These solutions span the different trade-offs between maximizing utilization for efficiency and maximizing isolation for availability, while enabling the system to be managed at a large scale and up and down

the layers of the systems architecture. Because that systems perspective informs everything we do, we have a wealth of solutions to match the diverse needs of modern enterprise architectures. Many of these tools are interoperable, enabling solutions that are otherwise impossible or impractical. Oracle's acquisition of Sun provides two further benefits to that portfolio: a secure future for these technologies and the exciting potential for integration with Oracle VM, Oracle Enterprise Manager, and the wealth of Oracle applications.

Here are some examples from the Sun portfolio. ZFS is a key storage virtualization technology at the core of the future of the Solaris operating system as well as the appliance products we build from Solaris technology today. Solaris networking virtualization technologies allow cutting-edge network hardware to be exploited and managed efficiently while providing a natural virtual network interface abstraction. For server virtualization, Solaris Zones (also known as Solaris Containers) have turned out to be very popular and very successful—a natural fit for the needs of many customers. The Logical Domains hypervisor is an extremely efficient design, and enables customers to get the most out of the tremendous throughput capability of SPARC CMT platforms. Our work with the Xen community enables a high-performance Solaris x64 guest for Oracle VM. For client virtualization, look no further than VirtualBox—for the laptop and desktop, both as a developer utility, and as a virtual appliance developer tool for the cloud. And it's not just a client technology: VirtualBox is the server component of Sun's virtual desktop infrastructure product, and VirtualBox continues to grow more server-class features with every release. As well as infrastructure virtualization platforms, we have created infrastructure management software—Ops Center—intended to reduce the complexity that comes with using the new capabilities in large-scale deployments.

Let's remember that virtual machines of one form or another have been around for a long time. Yet virtualization is such a fundamental idea that it remains associated with many developing fields. In the past decade, the runaway success of hypervisor-based virtualization on x64 platforms has largely been driven by the operational savings achieved by consolidating Microsoft Windows guests. But now this layer of the system architecture is just part of the way infrastructure is done, a new raft of capabilities can be built on top of it.

Recently we've seen the emergence of the Infrastructure as a Service (IaaS) style of cloud computing. Enabled by the combination of ever-increasing Internet connectivity and bandwidth, coupled with Moore's law providing more and more computational power per dollar, users of an IaaS service send their entire software stacks to remote data centers. Virtualization decouples the software from the hardware to enable those data centers to be operated almost as a utility. This approach promises to revolutionize the fundamental economics across the IT industry. The capital expense currently spent on under-utilized equipment can be

shifted to pay-as-you-go operating expenses, both within large enterprises and between service providers and their customers.

This new layer of the systems architecture brings new opportunities and new problems to solve: security, observability, performance, networking, utilization, power management, migration, scheduling, manageability, and so on. While both industry and the academic research community are busily responding to many of those challenges, there is still much to be done. The fundamentals remain important, and will continue to differentiate the various virtualization solutions in the marketplace.

And so I'm confident that there's yet more innovation to come.

This book is a deep exploration of virtualization products and technologies provided by or for Solaris, written by experienced practitioners in the art of delivering real solutions to data center problems. The book provides a holistic view of virtualization, encompassing all of the different models used in the industry. That itself is rare: No other organization has as complete a view of the entire range of system virtualization possibilities. A comprehensive background chapter leads neophytes into virtualization. Experienced data center architects will appreciate the individual chapters explaining the technologies and how you can use them to solve real problems—a critical resource in a rapidly changing world. I hope you find it as fascinating as I do!

Tim Marsland  
Vice President and Fellow, Sun Microsystems, Inc.  
Menlo Park  
February 18, 2010

*This page intentionally left blank*



# Preface

---

Computer virtualization has become its own sub-industry, with predictions that virtualization software and services revenues will exceed \$10 billion annually in 2011. Although numerous corporations use some form of computer virtualization, today that usage may be limited to only a small portion of their computers. As the number of virtualization options increases, however, the types of computers that can take advantage of the virtualization and the benefits provided by virtualization will increase.

*Oracle® Solaris 10 System Virtualization Essentials* is part of a new series of books on Oracle Solaris system administration. It presents the multiple technologies that the Oracle Solaris operating system uses to virtualize and consolidate computing resources, from hardware partitioning to virtual machines and hypervisors to operating system virtualization.

The intent of *Oracle Solaris 10 System Virtualization Essentials* is to discuss computer virtualization in general and to focus on those system virtualization technologies provided by, or that provide support to, the Oracle Solaris or OpenSolaris operating systems.

Oracle Solaris 10 supports a rich collection of virtualization technologies:

- Dynamic Domains
- Oracle VM Server for SPARC (previously called Sun Logical Domains)
- Oracle VM Server for x86

- Oracle VM VirtualBox (previously called VirtualBox)
- Oracle Solaris Containers (also called Zones)

Virtualization offers a tremendous opportunity to add computing workloads while controlling operational costs and adding computing flexibility. For the system administrator, this new knowledge area requires skills with new technologies like hypervisors, which create virtual machines on a single hardware machine, and containers (also known as zones), which create virtual operating systems running on a single operating system.

*Oracle Solaris 10 System Virtualization Essentials* describes the factors that affect your choice of technologies and explains how to

- Use Dynamic Domains to maximize workload isolation on Sun SPARC systems
- Use Oracle VM Server for SPARC to deploy different Oracle Solaris 10 and OpenSolaris environments on SPARC CMT (chip multithreading) systems
- Use Oracle VM Server for x86 or xVM hypervisor to deploy a server with heterogeneous operating systems
- Use Oracle VM VirtualBox to develop and test software in heterogeneous environments
- Use Oracle Solaris Containers to maximize efficiency and scalability of workloads
- Use Oracle Solaris Containers to migrate Solaris 8 and Solaris 9 workloads to new hardware systems
- Mix virtualization technologies so as to maximize workload density

*Oracle Solaris 10 System Virtualization Essentials* contains nine chapters. Chapter 1 discusses system virtualization in general terms. This material includes the needs of consolidation, the value and benefits of virtualization, and a description of the most common types of computer virtualization. Along the way, Chapter 1 also describes many of the concepts, features, and methods shared by many implementations of system virtualization. The concepts introduced in Chapter 1 are included in all of the other chapters.

Chapters 2 through 6 describe Oracle's computer virtualization technologies that are directly related to the Oracle Solaris OS, including their relationship with the topics introduced in Chapter 1. Chapter 7 discusses the factors that should be considered when choosing a virtualization technology or combination of technologies. That chapter also details a process of analysis that can be used to choose a virtualization technology or combination of technologies. Chapter 8 walks you through several examples of those technologies, and Chapter 9 completes the

picture by describing virtualization management software. Finally, the Appendix offers a narrated tour of the history of virtualization.

Because this book focuses on system virtualization technologies, technologies and methods that do not virtualize a computer system are not discussed. These include storage virtualization and application virtualization.

## **Books in the Oracle Solaris System Administration Series**

The *Oracle Solaris System Administration Series* includes the following books:

### **Solaris 10 System Administration Essentials**

*Solaris 10 System Administration Essentials* covers all of the breakthrough features of the Oracle Solaris 10 operating system in one place. It does so in a straightforward way that makes an enterprise-level operating system accessible to system administrators at all levels.

*Solaris 10 System Administration Essentials* provides a comprehensive overview, along with hands-on examples, of both the key features that have made Oracle Solaris the leading UNIX operating system and the significant new features of Solaris 10 that put it far ahead of its competitors. Features covered include zones, the ZFS file system, Fault Management Architecture, Service Management Facility, and DTrace (the dynamic tracing tool for troubleshooting OS and application problems on production systems in real time).

### **Solaris 10 Security Essentials**

*Solaris 10 Security Essentials* covers all of the security features and technologies in Oracle Solaris 10 that make it the OS of choice for IT environments that require optimal security.

*Solaris 10 Security Essentials* explains the strengths of Solaris security, including its scalability and adaptability, in a simple, straightforward way. It describes how security features in Oracle Solaris can protect a single-user system with login authentication as well as how those features can protect Internet and intranet configurations.

### **Solaris 10 ZFS Essentials**

*Solaris 10 ZFS Essentials* describes the dramatic advances in data management introduced by ZFS. ZFS provides an innovative approach to data integrity, near zero administration, and a welcome integration of file system and volume management capabilities.

*Solaris 10 ZFS Essentials* explains how to set up, configure, administer, and manage ZFS file systems, including how to install and boot ZFS as a root file system. It covers managing pools, configuring ZFS snapshots, and sharing ZFS home directories. It also illustrates a lab setup in a virtual machine that can be created on a laptop for a complete experimental environment.

## Intended Audience

The books in the *Oracle Solaris System Administration Series* can benefit anyone who wants to learn more about Oracle Solaris 10. They are written to be particularly accessible to system administrators who are new to Solaris—people who are perhaps already serving as administrators of Linux, Windows, or other UNIX systems.

If you are not presently a practicing system administrator but want to become one, this series, starting with *Solaris 10 System Administration Essentials*, provides an excellent introduction to this field. In fact, most of the examples used in the books are suited to or can be adapted to small learning environments such as a home system. Thus, even before you venture into corporate system administration or deploy Oracle Solaris 10 in your existing IT installation, these books will help you experiment in a small test environment.

*Oracle Solaris 10 System Virtualization Essentials* is especially valuable to several specific audiences. A primary group is generalists who desire knowledge of the entire system virtualization space. The only assumed knowledge is general UNIX or Linux administrative experience. Another group is data center staff who need an understanding of virtualization and use of such technologies in real-world situations.

- **Data center architects** will benefit from the broad coverage of virtualization models and technologies, enabling them to optimize system and network architectures that employ virtualization. The extensive coverage of resource controls can lead to better stability and more consistent performance of workloads in virtualized systems.
- **Computer science students** with UNIX or Linux experience will gain a holistic understanding of the history and current state of the system virtualization industry. The breadth of virtualization models discussed provides a framework for further discovery, and the real-world examples prepare students for data center careers.
- **Technical support staff** who troubleshoot virtualized systems will gain an introduction to system virtualization and interactions between virtualized systems. This background can shorten the time to diagnose problems, and enable personnel to readily distinguish between problems related to virtualization and ones that are independent of virtualization.



## How to Use This Book

Readers who wish to learn about one specific Oracle Solaris virtualization technology should read Chapter 1 and the appropriate sections of Chapters 2 through 6, 8, and 9. If you would like to understand all of the virtualization technologies that use Oracle Solaris as a core component and determine how to choose among them, read all of the chapters in this book. If you already understand virtualization but want to learn about virtualization using Oracle Solaris or OpenSolaris, you should skim through Chapter 1 to understand the context of the rest of the book as well as the definitions of terms used throughout the book. If you are implementing virtualization technologies on many systems, you should read Chapter 9 to understand the unique problems that must be addressed as part of this work and to identify software that can significantly reduce the complexity of large virtualization farms.

### Note

Oracle Corporation acquired Sun Microsystems, Inc., early in 2010, when this book was nearing completion. Although this book mostly uses the new product names, occasional reference is made to previous names. The following table provides a guide to the old and new product names.

Sun Product Name	Oracle Product Name
Solaris	Oracle Solaris
Solaris Containers, Containers, or zones	Oracle Solaris Containers, Containers, or zones
Logical Domains	Oracle VM Server for SPARC
Oracle VM	Oracle VM Server for x86
VirtualBox	Oracle VM VirtualBox

## OpenSolaris

In June 2005, Sun Microsystems introduced OpenSolaris, a fully functional release of the Solaris operating system built from open-source software. Although the books in this series focus on Oracle Solaris 10, they often incorporate aspects of OpenSolaris. Since that time, the evolution of this OS has accelerated even beyond its normally rapid pace. The authors of this series have often found it interesting to introduce features or nuances that are new in OpenSolaris. At the same, many of the enhancements introduced into OpenSolaris are finding their way into Oracle Solaris. So whether you are learning Oracle Solaris 10 or already have your eye on OpenSolaris, the books in this series are for you.

# Oracle VM VirtualBox

---

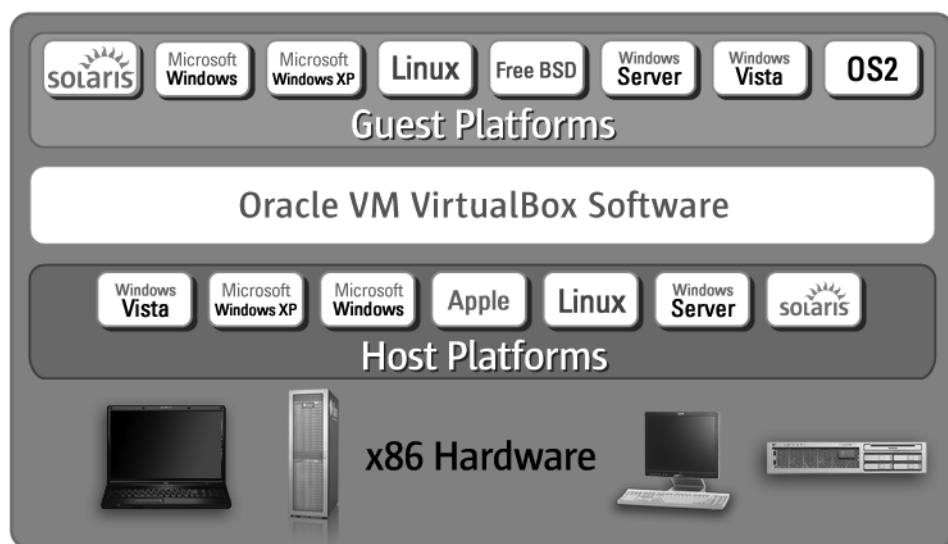
*Oracle VM VirtualBox (“VirtualBox”) is a high-performance, cross-platform virtualization engine for use on computers running Microsoft Windows, the most popular Linux distributions, Oracle Solaris, or MacOS. Designed for use on Intel and AMD x86 systems, Oracle VM VirtualBox can be deployed on desktop or server hardware. As a hosted hypervisor, it extends the existing operating system installed on the hardware rather than replacing it.*

*VirtualBox includes a hypervisor for the host platform, an application programming interface (API) and software development kit (SDK) for managing guest virtual machines, a command-line tool for managing guests locally, a web service for remote management of guests, a wizard-style graphical tool to manage guests, a graphical console for displaying guest applications on the local host, and a built-in Remote Desktop Protocol (RDP) server that provides complete access to a guest from a remote client.*

As shown in Figure 5.1, VirtualBox can run on a wide variety of host platforms. Binaries are available for these operating systems, most of them in 32-bit and 64-bit versions:

- Solaris 10 5/08 and newer, and OpenSolaris 2008.05 and newer
- Oracle Enterprise Linux (32-bit)
- Microsoft Windows (XP, Vista, 7) and Windows Server 2003 and 2008
- Mac OS X 10.5 and newer (Intel only)
- Linux distributions, including SuSE 9 and newer, Ubuntu, Red Hat Enterprise Linux 4 and newer, and others

There are no specific limitations on the guest operating system, but supported guests include all of the host operating systems plus FreeBSD, OS/2, and legacy Windows versions (NT, Windows 98, Windows 3.1, DOS). No special hardware is required to run VirtualBox, other than an Intel x86-compatible system and adequate memory to run the guests. If the system has Intel VT-x or AMD-V hardware virtualization extensions and they are enabled in the BIOS, VirtualBox can take advantage of these items and provide even better guest operational behavior.



**Figure 5.1** Platforms Supported by Oracle VM VirtualBox

The modular design of VirtualBox provides a consistent set of features across a wide range of host platforms. As a consequence, a virtual machine or disk image created on one host can be loaded and run on any supported host. In addition, a user or administrator who is familiar with managing guest virtual machines on one type of host can manage guests on any of the other supported systems.

Advanced desktop features such as Seamless Mode and Shared Clipboard give users a uniquely intimate experience when interacting with locally running guests. The built-in Remote Desktop Protocol (RDP) server makes VirtualBox ideal for consolidating and hosting remote desktop systems. Recent improvements in disk and network performance, especially when combined with the advanced resource management features available in Oracle Solaris, make VirtualBox an excellent choice for hosting server workloads.

This chapter assumes general knowledge of PC hardware. It also assumes the use of VirtualBox version 3.1.4.

## 5.1 How Oracle VM VirtualBox Works

Virtualizing an operating system on an x86 processor is a difficult task, especially without Intel VT-x or AMD-V hardware features. Before describing how VirtualBox works, a quick review of the x86 storage protection model is necessary.

The Intel x86 architecture defines four levels of storage protection called *rings*, which are numbered from 0 (the most privileged) to 3 (the least privileged). These rings are used by operating systems to protect critical system memory from programming errors in less-privileged user applications. Of these four levels, ring 0 is special in that it allows software to access real processor resources such as registers, page tables, and service interrupts. Most operating systems execute user programs in ring 3 and their kernel services in ring 0.

VirtualBox runs a single process on the host operating system for each virtual guest. All of the guest user code is run natively in ring 3, just as it would be if it were running in the host. As a result, user code will perform at native speed when running in a guest virtual machine.

To protect the host against failures in the guest, the guest kernel code is not allowed to run in ring 0 but instead runs in ring 1 if there is no hardware virtualization support, or in a VT-x ring 0 context if such support is available. This presents a problem because the guest may be executing instructions that are permitted only in ring 0 while other instructions behave differently when run in ring 1. To maintain proper operation of the guest kernel, the VirtualBox Virtual Machine Monitor (VMM) scans the ring 1 code and either replaces the troublesome code paths with direct hypervisor calls or executes them in a safe emulator.

In some situations, the VMM may not be able to determine exactly what the relocated ring 1 guest code is doing. In these cases, VirtualBox makes use of a QEMU emulator to achieve the same general goals. Examples include running BIOS code, real-mode operations early during guest booting when the guest disables interrupts, or when an instruction is known to cause a trap that may require emulation.

Because this emulation is slow compared to the direct execution of guest code, the VMM includes a code scanner that is unique for each supported guest. As mentioned earlier, this scanner will identify code paths and replace them with direct calls into the hypervisor for a more correct and efficient implementation of the operation. In addition, each time a guest fault occurs, the VMM will analyze the cause of the fault to see if the offending code stream can be replaced by a less expensive method in the future. As a consequence of this approach, VirtualBox performs better than a typical emulator or code recompiler. It can also run a fully virtualized guest at nearly the same speed as one that is assisted by Intel VT-x or AMD-V features.

Some operating systems may run device drivers in ring 1, which can cause a conflict with the relocated guest kernel code. These types of guests will require hardware virtualization.

5.1.1 Oracle VM VirtualBox Architecture

VirtualBox uses a layered architecture consisting of a set of kernel modules for running virtual machines, an API for managing the guests, and a set of user programs and services. At the core is the hypervisor, implemented as a ring 0 (privileged) kernel service. Figure 5.2 shows the relationships between all of these components. The kernel service consists of a device driver named `vboxsrv`, which is responsible for tasks such as allocating physical memory for the guest virtual machine, and several loadable hypervisor modules for things like saving and restoring the guest process context when a host interrupt occurs, turning control over to the guest OS to begin execution, and deciding when VT-x or AMD-V events need to be handled.

The hypervisor does not get involved with the details of the guest operating system scheduling. Instead, those tasks are handled completely by the guest during its execution. The entire guest is run as a single process on the host system and will run only when scheduled by the host. If they are present, an administrator can use host resource controls such as scheduling classes and CPU caps or reservations to give very predictable execution of the guest machine.

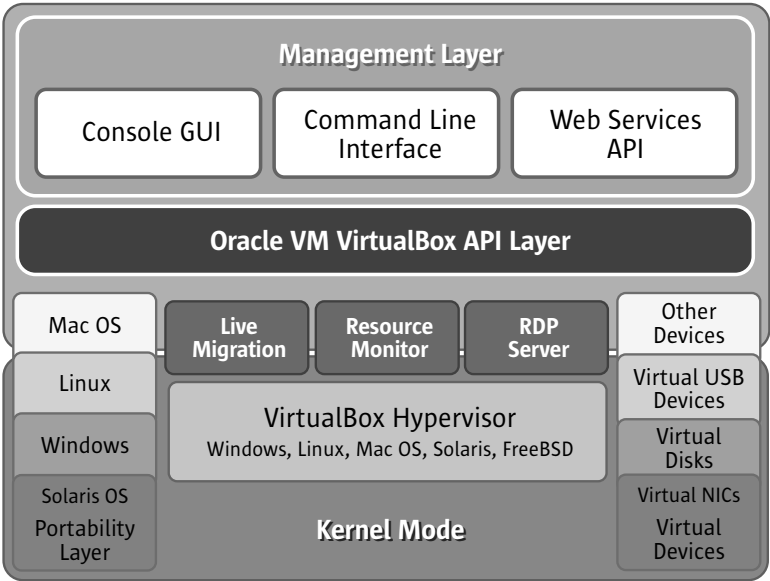


Figure 5.2 Oracle VM VirtualBox Architecture

Additional device drivers will be present to allow the guest machine access to other host resources such as disks, network controllers, and audio and USB devices. In reality, the hypervisor actually does little work. Rather, most of the interesting work in running the guest machine is done in the guest process. Thus the host's resource controls and scheduling methods can be used to control the guest machine behavior.

In addition to the kernel modules, several processes on the host are used to support running guests. All of these processes are started automatically when needed.

- `VBoxSVC` is the VirtualBox service process. It keeps track of all virtual machines that are running on the host. It is started automatically when the first guest boots.
- `vboxzoneaccess` is a daemon unique to Solaris that allows the VirtualBox device to be accessed from an Oracle Solaris Container.
- `VBoxXPCOMIPCD` is the `XPCOM` process used on non-Windows hosts for inter-process communication between guests and the management applications. On Windows hosts, the native COM services are used.
- `VirtualBox` is the process that actually runs the guest virtual machine when started. One of these processes exists for every guest that is running on the host. If host resource limits are desired for the guest, this process enforces those controls.

### 5.1.2 Interacting with Oracle VM VirtualBox

There are two primary methods for a user to interact with VirtualBox: a simple graphical user interface (GUI) and a very complete and detailed command-line interface (CLI). The GUI allows the user to create and manage guest virtual machines as well as set most of the common configuration options. When a guest machine is started from this user interface, a graphical console window opens on the host that allows the user to interact with the guest as if it were running on real hardware. To start the graphical interface, type the command `VirtualBox` at any shell prompt. On Oracle Solaris, this command is found in `/usr/bin` and is available to all users.

The CLI is the `VBoxManage` command. `VBoxManage` has many subcommands and options, some of which are discussed in the following sections. To get a list of all `VBoxManage` options, just type `VBoxManage` at any shell prompt. Without any command arguments, `VBoxManage` will respond with a list of all valid options. When a `VBoxManage` command successfully completes, it will print out a banner similar to the one in the following example:

```
% VBoxManage list vms
Sun VirtualBox Command Line Management Interface Version 3.1.4
(C) 2005-2010 Sun Microsystems, Inc.
All rights reserved.

"Windows XP" {4ec5efdc-fa76-49bb-8562-7c2a0bac8282}
```

If the banner fails to print, an error occurred while processing the command. Usually, diagnostic information will be displayed instead of the banner. If the banner is the only output, the command successfully completed. In the examples in the remainder of this chapter, the banner output has been omitted for the sake of brevity.

## 5.2 Oracle VM VirtualBox Guest Platform

VirtualBox supports the execution of guest operating systems in fully virtualized machines—a capability that allows the guest to run without requiring any special software or device drivers. The guest operating system is presented with a virtual motherboard with the following features.

- 1 to 32 CPUs
- Up to 32 GB of memory
- A dual-channel IDE disk controller with up to four devices
- An optional Serial ATA (SATA) disk controller with up to 30 attached devices
- An optional SCSI controller with up to 16 attached devices
- Up to 8 PCI network host adapters
- Keyboard, video, and mouse (KVM) console
- Either a legacy BIOS or EFI firmware

The next several sections describe details of the VirtualBox guest platform.

### 5.2.1 Virtual CPUs

Unlike Oracle VM Server for SPARC (previously called Sun Logical Domains), VirtualBox does not directly assign CPU resources to the guest domain. Instead, virtual CPUs are presented to the guest and time-sliced on real CPUs using the host system's scheduling facilities. The number of CPUs allocated for each guest can be specified in the Processor tab of the guest machine's System settings, as shown in Figure 5.3.

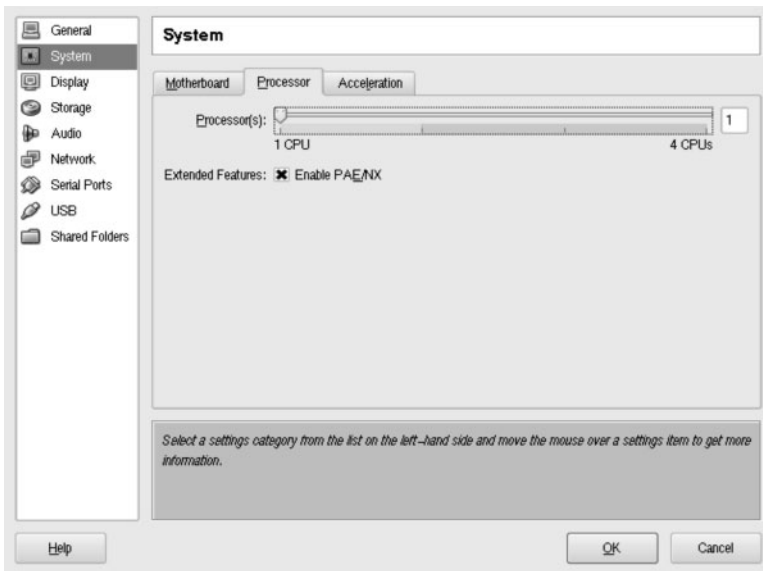


Figure 5.3 Configuring Virtual Processors

If you try to specify more CPUs than are present on the system, the VirtualBox graphical interface will display a warning. The guest will run but the performance of the guest is likely to be significantly degraded.

The number of CPUs can also be specified using the `VBoxManage modifyvm` command. `VBoxManage` will not issue a warning if the number of CPUs exceeds the capacity of the host.

```
% VBoxManage showvminfo "Solaris 10" | grep CPU
Number of CPUs: 1
CPUID overrides: None
% VBoxManage modifyvm "Solaris 10" --cpus 8
% VBoxManage showvminfo "Solaris 10" | grep CPU
Number of CPUs: 8
CPUID overrides: None
```

VirtualBox offers support for non-executable pages (NX). This feature enables guest operating systems to mark a page that is used for data so that it cannot be executed. This technique can help reduce the chance that a buffer overflow type of attack from a worm or virus against the guest will be successful. If the guest supports the NX feature, it is recommended that it be enabled in the CPU settings.

Although a guest does not require hardware virtualization assistance to perform well, if the host platform supports nested page tables, enabling this feature for a guest will provide a significant improvement in performance because most



of the memory management functions can be carried out by the guest without requiring host intervention. Nested page tables can be enabled in the Acceleration tab of the system settings in the VirtualBox GUI.

## 5.2.2 RAM

Unlike Logical Domains, memory used by guests is under the control of the host platform. Although the guests can take advantage of hardware virtualization features such as nested page tables to reduce the overhead of memory management, it is still possible to request more memory for a guest than the host system has available. If a shortfall of memory occurs due to this type of oversubscription, the host operating system will start demand paging, which may drag down the performance of the guest machines and other applications and services running on the host. VirtualBox does not yet provide a memory ballooning feature that would enable the host to take pages back from a guest in the event of a memory shortfall. Figure 5.4 shows an example of a guest whose memory allocation exceeds the recommended maximum.

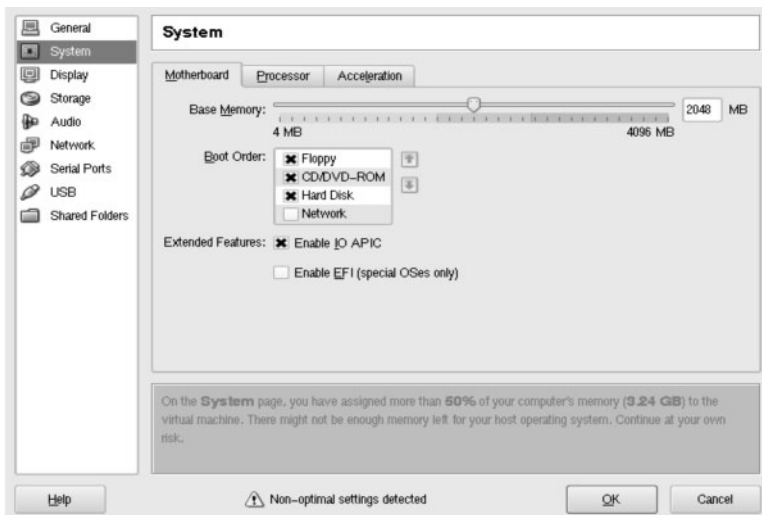


Figure 5.4 Setting Memory for an Oracle VM VirtualBox Guest

To help prevent overallocation of the host memory, the VirtualBox GUI will display a warning if it thinks that too much memory is being configured for a guest. In doing so, it considers only the total amount of memory on the system, not how much of that is actually free. It will display a warning if more than 50% of memory is allocated to a guest and a stronger warning if more than 80% is reserved for a guest.

If sufficient free memory is not available at the time, a guest machine may fail to start with an out-of-memory error, even if it will consume only a small amount of the memory allocated for its use.

Most guest systems run only a few applications or services. Because the host system performs most of the real operations on behalf of the guest, smaller memory allocations for the guest can still produce excellent results. Running a guest desktop with 512 MB or less may work well as long as enough memory is allocated for the guest system to boot.

### 5.2.3 Virtual Disk

Because a guest operating system needs persistent storage, VirtualBox can make several different types of host storage available to the guest in the form of virtual disks:

- A file that contains a disk image
- A real disk device
- An iSCSI target
- A CD-ROM/DVD or file containing an ISO image
- A file containing a floppy disk image

All of these storage options must be presented to the guest OS using a virtual disk. To facilitate this sharing, the VirtualBox virtual platform can have up to three disk controllers. Each of these controllers can have attached devices that provide access to the host storage.

Each virtual motherboard has a dual-channel IDE controller. Each of the two channels has two devices: a master and a slave. By convention, the master device on the first IDE channel is the boot disk and the master device on the second IDE channel is a CD-ROM/DVD. Many IDE chipsets are available, and VirtualBox can emulate an Intel PIIX3, PIIX4, or ICH6. There is no difference in the performance between these options, but if an operating system is expecting a particular IDE controller and sees a different one, it may not operate properly. This situation happens most often when importing a virtual machine from another virtualization product. To prevent this problem, set the IDE controller type to match that of the other virtualization product.

A virtual motherboard may also have a Serial ATA (SATA) controller. Such a controller can support up to 30 disk devices. By default, the first 4 devices operate in legacy IDE mode, meaning that the BIOS can use them just like any other IDE device. Once the guest operating system is up and running and has loaded the SATA drivers for these devices, they can then be accessed in SATA mode. In addition to supporting a larger number of devices, SATA is a more efficient interface both on the guest and in the emulation layer. SATA devices are preferred if the operating system supports them.

VirtualBox can also provide an LSI Logic or BusLogic SCSI controller, if necessary. Such a controller supports up to 16 devices. It is intended to facilitate use of legacy operating systems that do not support SATA and need more than the 4 devices provided by the IDE controller. This controller can also be used to attach more than the 30 disks supported by the SATA controller.

Guest hard disks are generally mapped to files on the host platform that contain a complete image of the guest disk, including the boot sector and partition table. The disk images have a fixed geometry based on their total size. Once the disk image is created, its size cannot be altered. When a guest reads from or writes to the disk, VirtualBox redirects the I/O to the native file system services on the host.

VirtualBox supports four disk image file formats.

- VDI, the native VirtualBox disk format. It is the default when you create a new virtual machine or disk image.
- VMDK, a popular disk format used by VMware.
- VHD, the format used by Microsoft.
- Parallels version 2 HDD format. VirtualBox does not support newer formats, but those can be converted to version 2 using tools supplied by Parallels.

With each of these formats, VirtualBox can create fixed-size or dynamically expanding disk images. Fixed-size image files are completely allocated at creation time. This type of image file will take longer to create, because it is dependent on the write performance of the host file system. Once in use, it will be more efficient, as the system does not need to get new blocks as the guest writes to new storage areas. In contrast, dynamically expanding disk images start off small and will grow as the guest writes to new blocks on the virtual disk. These are faster to create, but additional work is required by the host to find new blocks the first time a guest accesses a particular part of the disk. Host file system caching strategies can hide most of the difference in performance, especially on a host that is not heavily loaded. For performance-critical applications that perform many disk writes, fixed-size disk images are recommended. For all other uses, the convenience of dynamically allocated images makes this approach the preferred method.

VirtualBox maintains a library of disk, CD-ROM, and floppy disk images. Before a disk or CD-ROM image can be used by a guest, it must be registered in the Virtual Media Manager. This can be done in the VirtualBox GUI or via the `VBoxManage openmedium` command. Once an image is registered, it can be assigned to an open port on any guest. Although a disk image may be connected to more than one guest, it can be used by only one guest at a time. A guest will fail to start if one of its disk images is connected to another guest that is currently running.

Using the `VBoxManage` command line, the following example creates a 16 GB dynamically expanding disk image and attaches it to port 3 of the SATA controller in the guest named Windows 7.

```
% VBoxManage createhd --filename /vbox/HardDisks/Windows7-user.vdi \
    --size 16000 --format VDI --variant Standard --remember

0%...10%...20%...30%...40%...50%...60%...70%...80%...90%...100%

Disk image created. UUID: 4a0ef971-13d1-428b-aded-5f8720155e0a

% VBoxManage showhdinfo 4a0ef971-13d1-428b-aded-5f8720155e0a
UUID:                4a0ef971-13d1-428b-aded-5f8720155e0a
Accessible:          yes
Description:
Logical size:         16000 MBytes
Current size on disk: 0 MBytes
Type:                 normal (base)
Storage format:       VDI
Location:             /vbox/HardDisks/Windows7-user.vdi

% VBoxManage storageattach "Windows 7" --storagectl "SATA Controller" \
    --port 3 --device 0 --type hdd \
    --medium /vbox/HardDisks/Windows7-user.vdi
```

The results of this command can be seen in the Storage settings of this Windows 7 guest, as shown in Figure 5.5.

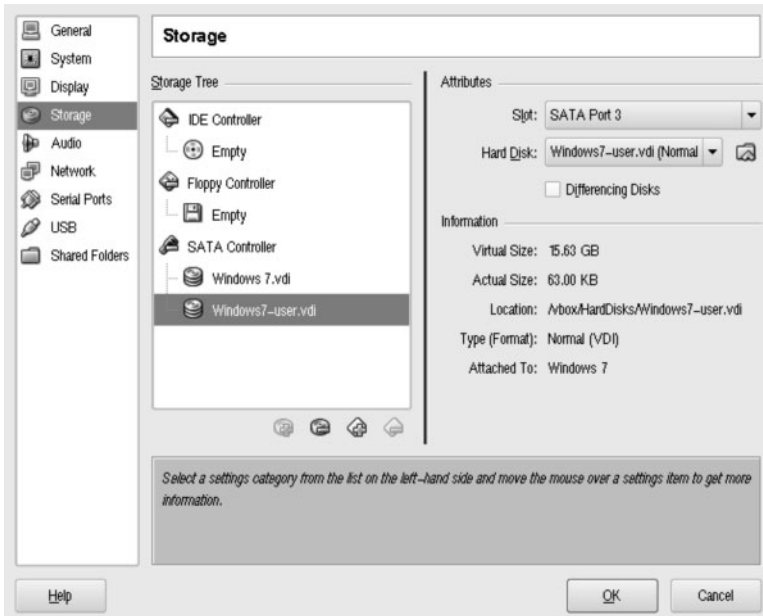


Figure 5.5 Selecting a Guest Disk Image

Note that the actual size of the disk image is only 63KB. To create a fixed-size image, add `--variant Fixed` to the `createhd` step in the preceding example.

CD-ROM images are treated in a similar fashion. The Virtual Media Manager maintains a list of registered images. Because CD-ROM images are not writable by the guest, they can be used by more than one guest at a time. One special image is `VBoxGuestAdditions.iso`, which can be found in the directory `/opt/VirtualBox/additions`. It contains all of the guest drivers and configuration tools that match the version of VirtualBox installed on the host. This image is automatically added by the installation program and is available to all guests.

In addition to a CD-ROM image, a guest can access real media in a CD-ROM or DVD drive on the host. By default, the guest is only allowed to read from the drive. To give the guest write access or to allow special I/O operations required by some multimedia players, enable the Passthrough setting on the CD-ROM device.

### 5.2.4 Virtual Network Devices

VirtualBox provides up to eight Ethernet PCI devices to each guest virtual machine. The user can select the type of virtual devices that are presented to the guest as well as what the host will do with the associated network I/O. The guest adapter does not need to be the same type as that on the host. For example, a Realtek Ethernet interface on the host can be virtualized as an Intel PRO/1000 on a guest. The first four virtual network adapters can be configured using the Network settings in the GUI. All eight of the devices can be configured using the `VBoxManage` command.

VirtualBox can present any of the following virtual devices to the guest operating system.

- **AMD PCNet PCI II:** A legacy host adapter for older guest operating systems.
- **AMD PCNet FAST III:** The default for most guests. This host adapter is well supported in most OS installation media, which makes it a good choice for most guests. It is also supported by the GNU GRUB bootloader, which allows network booting and installation of a guest operating system.
- **Intel PRO/1000 MT Desktop:** The default for newer guest operating systems such as Windows Vista and Windows Server 2008. Use this where the PCNet adapter is no longer supported or available on the installation media.
- **Intel PRO/1000 T Server:** Specifically for use with Windows XP guests.
- **Intel PRO/1000 MT Server:** A driver that allows guests to be imported from other virtualization products such as VMware. The PRO/1000 MT Server virtual device is commonly used on those platforms.
- **Virt-io:** A device used for guests that support a KVM paravirtualized (PV) network interface. Because this device is designed for virtualization, it may

offer performance advantages over the other emulated devices. PV drivers are available in 2.6.25 or later Linux kernels. They can be downloaded for other operating systems at the KVM project page.

In addition to choosing the virtual device for the guest, the user must configure a host networking mode to use for each device. Five different modes are available, each of which offers some interesting benefits.

*Not attached* is a mode similar to a network adapter that doesn't have a network cable attached. The device is present in the guest machine but is not reporting a positive link status. Traffic will not flow through this device in this mode.

*Network Address Translation* (NAT) will hide the guest's interface behind a network tunnel. This mode is often used when the guest is a desktop system and primarily a consumer of network resources rather than a provider. To assist guests that automatically detect their network settings, VirtualBox provides a DHCP server, router, and DNS proxy to resolve network names and correctly route packets. NAT has a few limitations that may cause applications to behave differently than they would on a real system. For example, ping may not work across the NAT tunnel. Some VPN products use this method to determine if a network is reachable, so these products would not work with a virtual NIC in NAT mode. In addition, jumbo frames are not reliable when using NAT.

Because external systems cannot communicate directly with a guest using NAT mode, VirtualBox can provide port redirection using the host's IP address. External systems connect to the specified port on the host; VirtualBox then redirects all of the packets to the guest. There are a few restrictions on the use of port forwarding. For example, you cannot redirect a port that is already in use by the host. Ports with numbers less than 1024 require the requester to be running as root or with the `net_privaddr` privilege. Because neither of these is a recommended practice for otherwise unprivileged users, you should choose a port on the host with a number greater than 1024. The most common use of NAT mode is to forward guest port 22, which allows an external system to access the guest using SSH. The following example establishes an SSH port from the first network adapter (an Intel Pro/1000 MT Desktop) on the guest named `Solaris 10` to port 2222 on the host:

```
% VBoxManage setextradata "Solaris 10" \  
    "VBoxInternal/Devices/e1000/0/LUN#0/Config/s10ssh/Protocol" TCP  
  
% VBoxManage setextradata "Solaris 10" \  
    "VBoxInternal/Devices/e1000/0/LUN#0/Config/s10ssh/GuestPort" 22  
  
% VBoxManage setextradata "Solaris 10" \  
    "VBoxInternal/Devices/e1000/0/LUN#0/Config/s10ssh/HostPort" 2222
```

The guest can be accessed by using `ssh -p 2222 user@host`.

*Bridged* is a more advanced network mode. When working in this mode, VirtualBox installs a software network that allows the guest to share a specific host interface. A randomly generated MAC address is assigned to the guest adapter and its full network stack is visible to external systems. All of the network operations are available to the guest, including ping and jumbo frames. The bridged mode is the recommended setting for guests running server applications and desktops requiring a VPN connection into another network.

*Internal mode* is used to communicate between virtual machines on the same host. It is similar to bridged mode except that all communications stay internal to the host platform. Traffic over the internal mode software network is also invisible to the host. Internal mode is the fastest and most secure method of communication between guests. The most common use for this mode is to establish a private secure channel for guests to share that cannot be observed by any external system or other applications on the host system. Examples include an internal NFS server or a content provider for an externally facing web service.

*Host only* mode is similar to *internal mode* except that the host is able to communicate with the guests. All communications are internal to the host, but applications and users on the host can observe and use network services on the guests.

Basic network settings for the first four adapters, as shown in Figure 5.6, can be specified on the Network settings screen in the VirtualBox graphical interface.

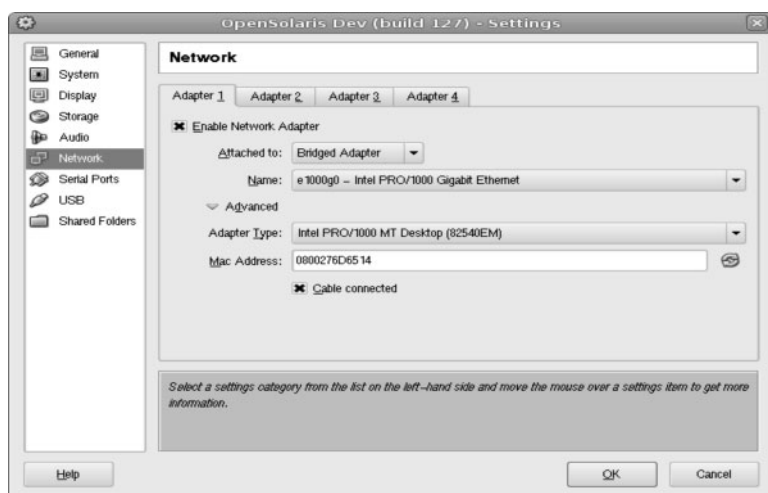


Figure 5.6 Oracle VM VirtualBox Guest Network Configuration Settings

### 5.2.5 BIOS and EFI

VirtualBox provides a virtual standard BIOS firmware that is used by the guest virtual machine during the boot process. Through the user interface, the user can select options such as boot order and support for I/O APIC.

VirtualBox also provides an Extended Firmware Interface (EFI) for operating systems such as Mac OS X that use EFI instead of the legacy BIOS. Newer versions of Windows and some Linux distributions can use either the legacy BIOS or EFI. The type of firmware is selected in the Motherboard part of the System settings. Figure 5.7 shows the BIOS and boot order settings for a guest machine.

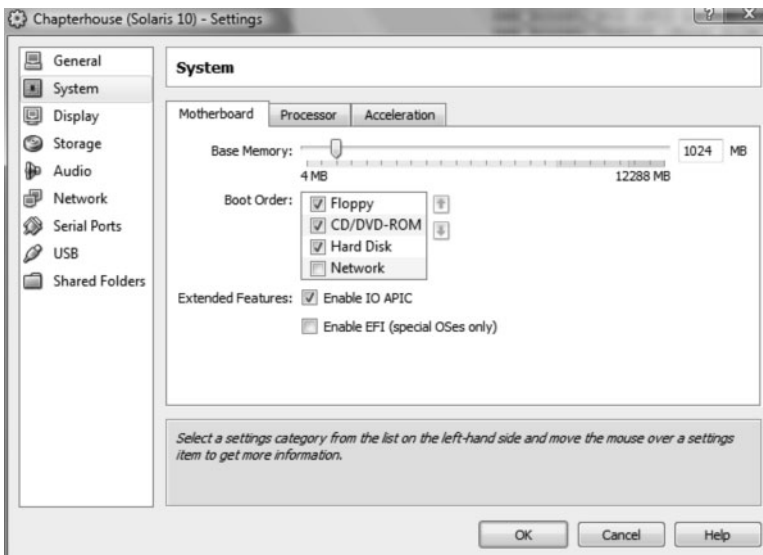


Figure 5.7 Selecting the Guest BIOS Type and Boot Order

You can also set the type of BIOS firmware by using the `VBoxManage modifyvm` command:

```
# VBoxManage modifyvm "Solaris 10" --firmware bios
```

These are just the basic features of the VirtualBox guest platform. For a complete list of all configuration options, see the *VirtualBox User Manual*, which is available in PDF format in the installation directory `/opt/VirtualBox/UserManual.pdf`. This document can be found at <http://www.virtualbox.org/wiki/Downloads>.



## 5.2.6 Guest Additions

Although guest operating systems do not need any special software or drivers to operate correctly, the host platform includes many features that a guest can utilize if they are available. Specifically, VirtualBox provides a special set of drivers and utilities that can be used by the guest once the operating system has been installed. These “Guest Additions” are provided in a CD-ROM disk image that is always available to the guest.

You can use one of two methods to install the Guest Additions. The first method is to attach the disk image file `VBoxGuestAdditions.iso` to one of the available virtual CD-ROM devices. The second method is to use the VirtualBox graphical console and choose `Devices → Install Guest Additions` from the guest console window. If the guest operating system does not automatically mount the media, this step can be done manually from the guest.

Once the virtual media has been mounted, run the appropriate installer for your guest platform. A reboot of the guest is required only if you are upgrading an older version of the guest additions that are already installed. For a new installation, all that is required is to log out of the guest and log back in again to use the Guest Additions.

Once installed, the Guest Additions provide several new features:

- **Integrated keyboard and mouse.** The guest no longer requires a captive keyboard and mouse. When the guest window receives focus, the keyboard and mouse are automatically attached. When the window loses focus, they are automatically released by the guest.
- **Shared clipboard.** This feature allows a user to copy an object from an application in one guest and paste it into another application on a totally different guest, or even on the host system.
- **Resize guest display.** This feature enables the user to resize the guest display. VirtualBox will notify the guest of the resolution changes, and the drivers inside the guest will handle the changes.
- **Seamless mode.** VirtualBox can hide the guest display background, only displaying application windows on the host. When used with the shared clipboard, this feature will hide the fact that applications are being run in virtual machines. Instead, all applications appear to be running together in a single environment—a uniquely integrated desktop experience.
- **Time synchronization.** This feature keeps the guest clock synchronized with that of the host, thereby avoiding the type of clock skewing that is common in virtualized environments.
- **High-performance graphics.** If this feature is enabled in the Display settings of the guest machine, VirtualBox will allow the guest operating

system to pass OpenGL and Direct3D graphics commands directly to the host's graphics adapter. The guest can then perform 3D graphics operations at nearly the same speed as if they were running on the host. For Windows guests, 2D graphics acceleration is also available.

- **Shared folders.** Shared folders allow the guest to access the host file systems as if they were local file systems. For Windows guests, these appear as network shares. For all other guests, a special file system driver is used to access the shared folders.

## 5.3 Oracle Solaris as an Oracle VM VirtualBox Host

VirtualBox supports Solaris 10 5/09 and later as a host platform with a few restrictions:

- There is no support for USB. VirtualBox uses a newer USB device management system that was introduced into OpenSolaris build 124 and is not available in Solaris 10.
- ACPI information, such as battery status and power source, is not reported to guest operating systems.
- Wireless network adapters cannot be used for bridged networks.

VirtualBox can run in either the global zone or a Solaris Container. Running in a Container provides several interesting benefits. It may be easier to place resource controls on an entire Container than an arbitrary workload, especially if different people are tasked with these functions. The global zone administrator can place resource policies on the Container that the VirtualBox user may not be aware of or have sufficient privilege to set. Migrating a Container by detaching it from one host and attaching it to another host can greatly simplify the task of relocating a guest. Most of the configuration settings needed for proper operation of the guest are contained in the Container configuration file, so any changes that need to be made can be done in one place. Finally, the ability to clone a Container that contains a guest and its associated data makes it easier to rapidly deploy several copies of the same machine on the host.

To enable VirtualBox operation in a Container, just add the device `/dev/vboxdrv` to the Container. For OpenSolaris hosts for which you want to enable USB support, add the device `/dev/vboxusbmon` in addition to `/dev/vboxdrv`. The following example shows the creation of a Container called `WinXP` that could be used to run VirtualBox guests. A more detailed example is provided in Chapter 8, “Applying Virtualization.”

```
# zonecfg -z WinXP
WinXP: No such zone configured
Use 'create' to begin configuring a new zone.
zonecfg:WinXP> create
zonecfg:WinXP> set zonepath=/zones/WinXP
zonecfg:WinXP> add device
zonecfg:WinXP:device> set match=/dev/vboxdrv
zonecfg:WinXP:device> end
zonecfg:WinXP> add net
zonecfg:WinXP:net> set physical=e1000g0
zonecfg:WinXP:net> set address=192.168.1.41/24
zonecfg:WinXP:net> end
zonecfg:WinXP> exit

# zoneadm -z WinXP install
Preparing to install zone <WinXP>.
Creating list of files to copy from the global zone.
Copying <35929> files to the zone.

# zoneadm -z WinXP boot
# zlogin -C WinXP
```

Finish the installation as you would any other Container by answering the Solaris system identification questions. Once this step is complete, you can use the VirtualBox graphic or command-line tools just as you would in the global zone. Some of the host networking modes may require the use of exclusive IP for configuration of the guest's virtual network interface.

Another Oracle Solaris feature that can be used by VirtualBox is ZFS. Although VirtualBox has a host-independent disk cloning feature, it works by copying entire disk images—an approach that doubles the amount of storage required for the duplicated clone and places a heavy load on the host system during the copying operation. By comparison, ZFS has a fast cloning capability that takes significantly less time to complete; the cloned disk image needs to store only those blocks that are different from the blocks in the original image. This strategy is very efficient for deploying many copies of the same type of guest.

When running VirtualBox on a Solaris host that is also using ZFS, limiting the size of the adaptive replacement cache (ARC) is recommended. By default, ZFS can use most of the physical memory on the system: up to three-fourths on systems with 4 GB or less and up to `maxphys-1` GB on larger systems. A guest may demand memory faster than ZFS is able to free it, which would produce one of the memory shortfall situations we are trying to prevent. The solution is to limit the amount of memory that ZFS is able to use for the ARC. This cap can be set with the `zfs_arc_max` Solaris tunable parameter, which you can set in `/etc/system`.

The following setting will limit the ARC to 1 GB, which is a reasonable value for a 4 GB desktop hosting several virtual machines:

```
set zfs:zfs_arc_max = 0x40000000
```

As a general rule, do not set `zfs_arc_max` equal to more than three-fourths of the physical memory that is left after allocating memory for all of your guests.

See Chapter 8, “Applying Virtualization,” for an example of using both ZFS and Containers for rapid provisioning of guests on an Oracle Solaris host.

### 5.3.1 Installing Oracle VM VirtualBox

The VirtualBox software can be downloaded from <http://virtualbox.org/downloads>. The Oracle Solaris version is provided in a single SVR4 data stream package that includes both the 32-bit and 64-bit versions of the software as well as a README text file describing the installation process and a package answer file named `autoresponse` for non-interactive installations. Because the package installation scripts load kernel modules, the installation must be done in the global zone and carried out either by root or by a user or role that has the Software Installation execution profile.

If an older version of VirtualBox is installed on the host, it must be removed before installing the new version. For versions prior to 3.1, two packages must be uninstalled: `SUNWvbox` and `SUNWvboxkern`. Starting with version 3.1, there is just a single package: `SUNWvbox`. It is not necessary to reboot the Solaris host after the old version of VirtualBox is removed or the new version is installed.

In the next example, a new version of VirtualBox is installed on a system that is already running an older release. Note the use of the included `autoresponse` file for unattended package operations.

```
# ls
VirtualBox-3.1.4-57640-SunOS.tar.gz

# /usr/sfw/bin/gtar xpfz VirtualBox-3.1.4-57640-SunOS.tar.gz

# ls
ReadMe.txt
VirtualBox-3.1.4-57640-SunOS.tar.gz
VirtualBox-3.1.4-SunOS-r57640.pkg
autoresponse

# pkgrm -n -a autoresponse SUNWvbox

Removing VirtualBox drivers and services...
- Unloaded: Web service
```

*continues*

```
- Unloaded: Zone access service
- Unloading: USB ...FAILED!
- Removed: USB module
- Unloaded: USBMonitor module
- Removed: USBMonitor module
- Unloaded: NetFilter module
- Removed: NetFilter module
- Unloaded: NetAdapter module
- Removed: NetAdapter module
- Unloaded: Host module
- Removed: Host module
Done.

# pkgadd -n -a autoreponse -d VirtualBox-3.1.4-SunOS-r57640.pkg all
Checking for older bits...
Installing new ones...

Loading VirtualBox kernel modules...
- Loaded: Host module
- Loaded: NetAdapter module
- Loaded: NetFilter module
- Loaded: USBMonitor module
- Loaded: USB module

Configuring services...
- Loaded: Web service
- Loaded: Zone access service

Installing Python bindings...
- Installed: Bindings for Python 2.4
- Installed: Bindings for Python 2.5
- Installed: Bindings for Python 2.6

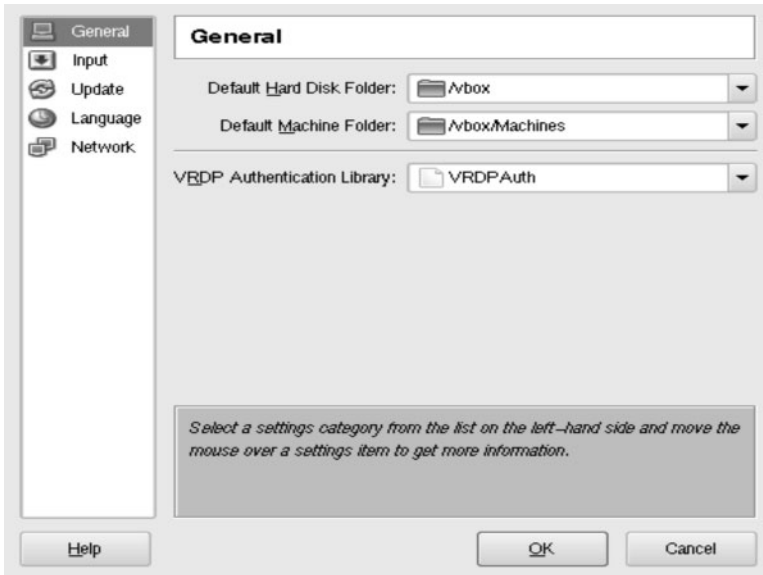
Updating the boot archive...

Installation of <SUNWvbox> was successful.
```

The default directory for the VirtualBox components is `/opt/VirtualBox`. The user commands `VirtualBox`, `VBoxManage`, `VBoxSDL`, `VBoxHeadless`, and `VBoxQtconfig` are all symbolically linked into `/usr/bin` so that they are available for all users on the system. In general, no special privileges are required to run VirtualBox on Solaris other than appropriate file permissions to devices and disk images. The privilege `net_priv_addr` would be required to forward a host port number less than 1024 to a guest, but this configuration is strongly discouraged.

When a user runs `VirtualBox`, all of the machine definitions and private disk images are stored by default in a directory named `.VirtualBox` in the user's

home directory. Although the machine configuration files are small, disk images can grow quite large. To change the locations where these files are stored, click **File → Preferences** in the VirtualBox graphical interface or use the `VBoxManage setproperty` command. Figure 5.8 shows how to change those preferences to a different location.



**Figure 5.8** Changing the Location of Guest Disks and Machine Definitions

VirtualBox disk images and machine configurations can be shared among many users. All that is required to share machine configurations and disk images is read and write file permissions to the associated files.

## 5.4 Oracle Solaris as an Oracle VM VirtualBox Guest

Solaris 10 is a fully supported guest OS for VirtualBox. A full complement of Guest Additions is available, including seamless mode and accelerated 3D graphics. For optimal performance, the following settings are recommended for a Solaris guest:

- Boot disk: SATA (one of the first four ports with IDE compatibility mode enabled)
- CD-ROM: Master device on the second IDE channel (the default)

- Network: Intel Pro/1000MT Desktop
- Hardware acceleration: Enabled if supported by the host
- Nested page tables: Enabled if supported by the host
- PAE/NX: Enabled

As with all supported guests, the Guest Additions are provided on a CD-ROM image file that is automatically installed with VirtualBox. The Oracle Solaris Guest Additions are included in a single SVR4 data stream package named `VBoxSolarisAdditions.pkg`. As with the host packages, if a previous release of the Guest Additions is installed, it must be removed before a new version can be installed. Rebooting the guest after the new additions are installed is strongly recommended; this step is not required when the Guest Additions are first installed.

The following command installs the Guest Additions on a new Solaris guest:

```
# pkgadd -d /cdrom/cdrom0/VBoxSolarisAdditions.pkg all
```

Once the Guest Additions are installed, all of the ancillary features—such as the ability to resize the guest display automatically and implement the shared clipboard—should be available for use.

One special feature of the Guest Additions is shared folders. It allows the guest to share files with other guests and the host via the host's native file system. In Oracle Solaris, the shared folders are made available as a `vboxfs` file system. Shared folders are defined per guest in the VirtualBox GUI or via the `VBoxManage` command line. In the following example, the directory `/export/iso` on the host is shared as `/iso` with a Solaris 10 guest. On the host platform, issue the following command to create the shared folder. In this example, the guest is named `Solaris10`.

```
Host% VBoxManage sharefolder add Solaris10 --name iso \
      --hostpath /export/iso
```

Now the guest can mount and access the file system, as in the following example:

```
# mkdir /iso
# mount -F vboxfs -o uid=1234,gid=5678 iso /iso
# ls -la /iso

total 19720801

drwxrwxrwx  1 1234      5678      4096 Dec  1 16:02 .
```

```

drwxr-xr-x 34 root      root          35 Feb 12 20:51 ..
drwxrwxrwx 1 1234      5678          4096 Sep  9 08:43 centos
drwxrwxrwx 1 1234      5678          4096 Aug 27 13:22 fedora
drwxrwxrwx 1 1234      5678          8192 Feb  1 12:20 opensolaris
drwxrwxrwx 1 1234      5678          4096 Oct 25 10:29 oracle
drwxrwxrwx 1 1234      5678          8192 Aug 31 13:44 redhat
drwxrwxrwx 1 1234      5678          4096 Sep  9 08:56 rescue
drwxrwxrwx 1 1234      5678          4096 Feb  3 16:12 s10
drwxrwxrwx 1 1234      5678          8192 Feb  3 21:57 s11
drwxrwxrwx 1 1234      5678           0 Aug 31 13:31 suse
drwxrwxrwx 1 1234      5678           0 Aug  9 20:09 ubuntu
drwxrwxrwx 1 1234      5678          8192 Feb 13 00:38 windows

```

Because the file permission and ownership abstractions may not translate directly between the host operating system and that of the guest, the user starting the virtual machine in the host must have appropriate access to the files being shared. Inside the guest, the owner and group are set by mount options—in this case, user 1234 and group 5678.

## 5.5 Creating and Managing Oracle VM VirtualBox Guests

After you have installed VirtualBox, you can create guests. While this can be accomplished using command-line tools, the first example given here uses the VirtualBox GUI installation wizard.

### 5.5.1 Creating the Guest Machine

You can start to install a guest machine by launching the VirtualBox tool:

```
% VirtualBox &
```

Figure 5.9 shows the initial VirtualBox screen.

To begin creating the first virtual machine, click the New button. This will launch the New Virtual Machine Wizard.

The first step is to name the new virtual machine and choose the operating system of the guest. The name of the guest is unrelated to its actual host name or network identity; those identities will be set later, once the guest is running. The guest name is only used by VirtualBox to identify the guest being managed. The operating system type determines how VirtualBox will emulate devices and which code scanning techniques to use when running the guest kernel code. The name of the guest can be changed later, but the OS type should be correct before creating the guest.



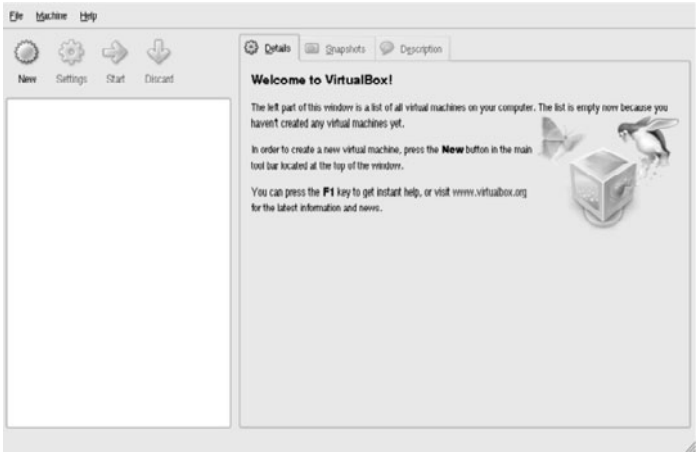


Figure 5.9 VirtualBox Main Window

For this example, the guest will be named `Windows XP` and will run the Windows XP operating system, as shown in Figure 5.10.



Figure 5.10 Choosing the Guest Operating System Type

The next step is to assign the amount of memory to be allocated for the guest. The amount needed is usually less than needed for a physical system. Many operating systems manage a file cache to improve performance of disk reads. If the host OS has a file cache, the guests will not need much of a file cache to maintain reasonable performance, so you can allocate less memory to guests. The installation wizard will suggest a minimum memory size based on the operating system type chosen in the previous step. Figure 5.11 shows the allocation of 512 MB for the Windows XP guest—more than enough for typical virtual desktop needs.



Figure 5.11 Setting the Amount of Guest Memory

Every guest needs a boot disk. For simplicity, the New Virtual Machine Wizard allows you to create only a single boot disk image. It will always assign this disk as the master device on the first IDE controller. If you need a different configuration, use the `VBoxManage` command. At this point in the guest installation, you can choose whether to create a new disk image or use one that is already registered in the Virtual Media Manager. There aren't any virtual disks on this system yet, so select Create new hard disk as shown in Figure 5.12. Note that VirtualBox suggests a size for the disk based on the operating system that was chosen in the previous step.



Figure 5.12 Virtual Hard Disk Wizard

This disk must either be a fixed size or dynamically expanding disk image. A dynamically expanding disk is typically used, as shown in Figure 5.13, because it doesn't waste real disk space.

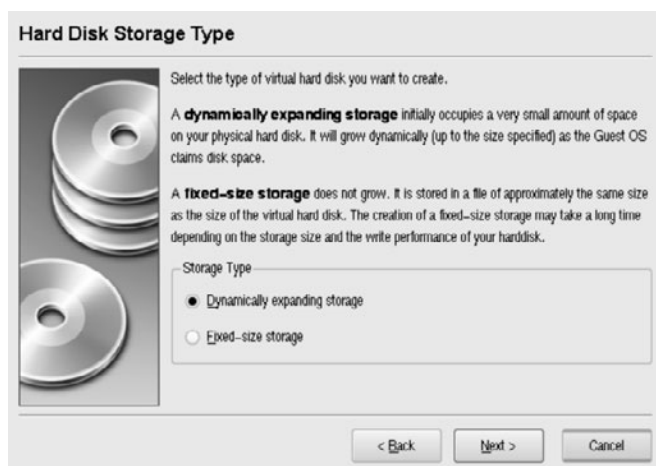


Figure 5.13 Choosing the Disk Image Type

The next screen allows you to name the disk image and specify its size. By default, this disk image is stored in your home directory in a hidden directory named `.VirtualBox`. If not enough space is available there, you can click the icon to the right of the image name and tell VirtualBox where to place the image. You will need read and write access to the directory holding the disk images.

In this example, the disk image is named `Windows 7.vdi` and has a size of 10 GB, which is the default. In practice, the name of the disk image should indicate not only which virtual machine is using it, but also how that disk is used. Figure 5.14 shows the name and size of the guest boot disk.

Figure 5.15 shows the final disk configuration. After reviewing the settings, click **Finish**. At this point, the disk image will be created and registered in the Virtual Media Manager.

The final step in the installation wizard is to review the final guest configuration, as shown in Figure 5.16. Verify that the settings are what you desired and click **Finish** to create the guest.

You have successfully created your first guest machine. You can click the **Settings** button and familiarize yourself with some of the other configuration options, such as audio, video memory size, additional networks, processor accelerations, and shared folders.



**Figure 5.14** Choosing the Name and Size of the Guest Disk Image (top)

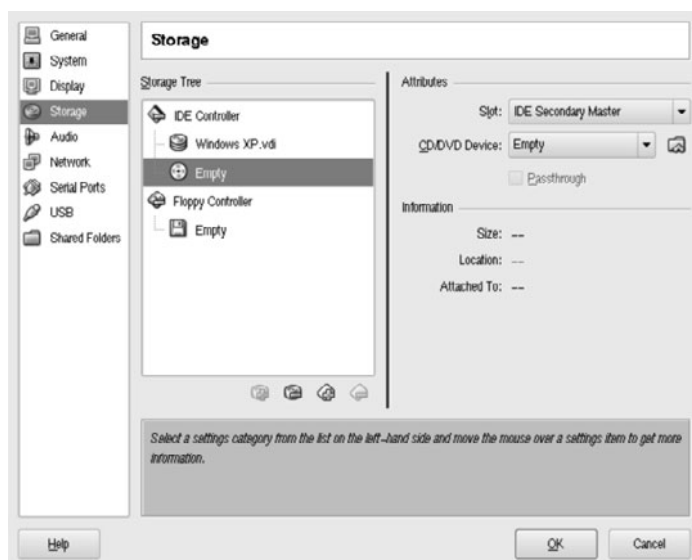
**Figure 5.15** Reviewing the Guest Disk Settings (center)

**Figure 5.16** Final Review of the Guest Configuration (bottom)

## 5.5.2 Installing the Guest Operating System

Once you are satisfied that the guest machine configuration is suitable for your needs, you can install the operating system on the guest boot disk. The most common method of doing so is to attach a CD-ROM disk image to one of the available virtual CD-ROM devices so that the guest will boot from that device. To attach the CD-ROM image, select the guest machine and click Settings. Then select Storage and a window similar to Figure 5.17 will be displayed.

When you click the CD-ROM device, you will see that it is the master device on the secondary IDE channel. This device also happens to be second device in the BIOS boot order, right behind the floppy disk, which is rarely used. Also note that the device is currently empty, meaning that no virtual or real media has been inserted.



**Figure 5.17** Guest Storage Configuration Before Attaching the Installation Media

To attach the installation media, you must first register it with the Virtual Media Manager. In this example, the media is found in the file `/export/iso/windows/winxp_sp3_x86.iso`. Click the folder icon to the right of the CD/DVD Device pull-down menu, which will launch the Virtual Media Manager, as shown in Figure 5.18.

Because this is the first guest in a new VirtualBox installation, the only CD-ROM disk image present is `VBoxGuestAdditions.iso`, which contains the Guest Additions. To add the Windows installation media, click the Add button, navigate

to the directory where the CD-ROM image is stored, and select it, as shown in Figure 5.19. Then you can highlight the appropriate disk image and click the Select button to insert it into the virtual CD-ROM device. Once registered, you can select this media in any guest from the CD/DVD Device pull-down menu on the guest Storage settings without having to start the Virtual Media Manager.

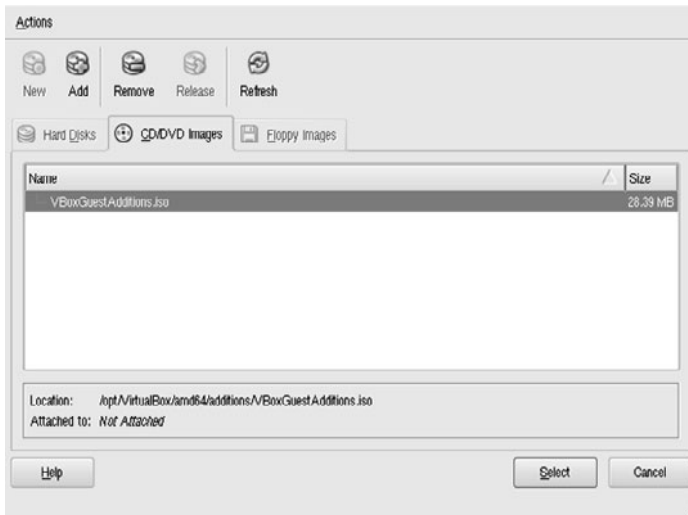


Figure 5.18 Virtual Media Manager

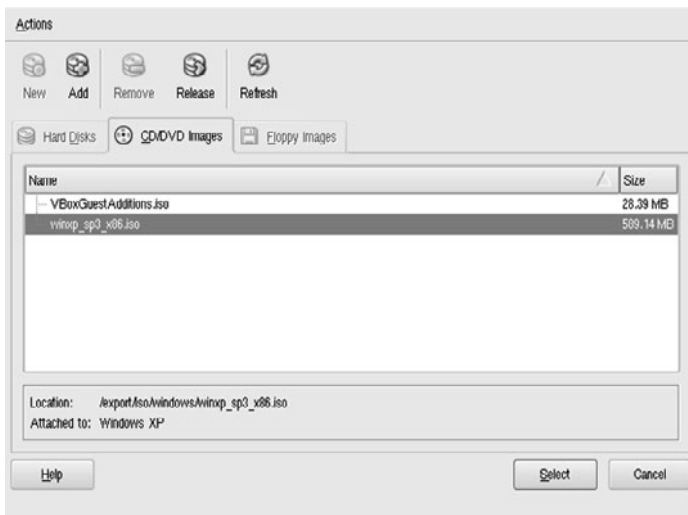
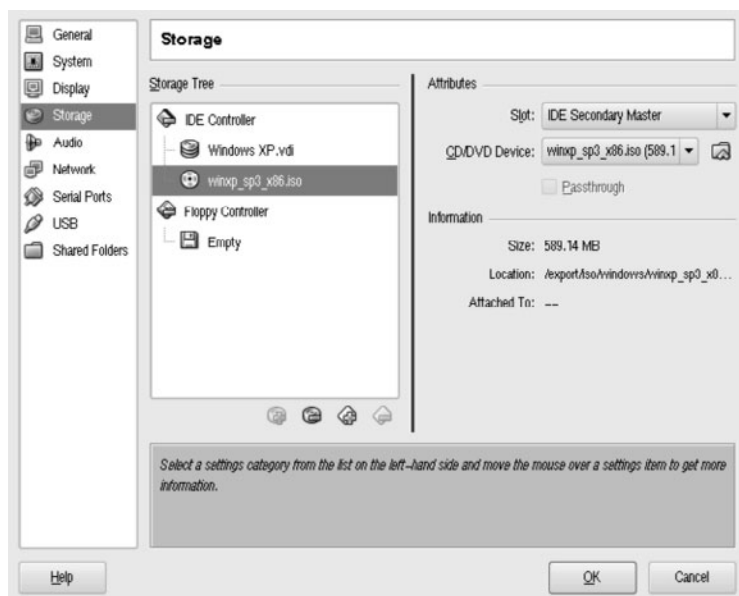


Figure 5.19 Virtual Media Manager After Adding the Windows Installation Media

After the CD-ROM disk image is registered with the Virtual Media Manager, it should be available as a CD-ROM device selection on the guest storage settings, as shown in Figure 5.20. Alternatively, if you start the guest with a new boot disk and fail to assign a CD-ROM, VirtualBox will launch the Virtual Media Manager so that you can select an existing image or register a new one. When this process is complete, the image will automatically be attached to the CD-ROM device and the guest will continue the booting process.



**Figure 5.20** Selecting the Installation Media for the Guest

All that is left to do is start the guest machine. However, first we will show the creation of a guest using the VBoxManage command line.

### 5.5.3 Creating a Guest Machine Using the Command Line

The command-line interface (CLI) for VirtualBox is more complicated to use than the GUI, but it offers several advantages:

- The CLI is more flexible than the GUI.
- The CLI offers features not available from the GUI.
- The CLI can be used to script guest creation.
- Scripts using the CLI can be used to build an automated guest creation system.

As in the previous example, the first step is to create and register the guest virtual machine. If you don't know which OS type to select when creating your virtual machine, giving the command `VBoxManage list ostypes` will produce a list of supported options. For this example, `WindowsXP_64` is the correct choice.

```
% VBoxManage createvm --name "Windows XP" \
--ostype WindowsXP_64 --register
Virtual machine 'Windows XP' is created and registered.
UUID: 4ec5efdc-fa76-49bb-8562-7c2a0bac8282

Settings file: '/vbox/Machines/Windows XP/Windows XP.xml'

% VBoxManage showvminfo "Windows XP"
Name:                Windows XP
Guest OS:             Windows XP (64 bit)
UUID:                4ec5efdc-fa76-49bb-8562-7c2a0bac8282
Config file:         /vbox/Machines/Windows XP/Windows XP.xml
Hardware UUID:       4ec5efdc-fa76-49bb-8562-7c2a0bac8282
Memory size:         128MB
VRAM size:           8MB
Number of CPUs:      1
Synthetic Cpu:       off
CPUID overrides:     None
Boot menu mode:      message and menu
Boot Device (1):     Floppy
Boot Device (2):     DVD
Boot Device (3):     HardDisk
Boot Device (4):     Not Assigned
ACPI:                on
IOAPIC:              on
PAE:                 on
Time offset:         0 ms
Hardw. virt.ext:     on
Hardw. virt.ext exclusive: on
Nested Paging:       on
VT-x VPID:           on
State:               powered off (since 2010-02-14T23:05:30.798000000)
Monitor count:       1
3D Acceleration:     off
Teleporter Enabled:   off
Teleporter Port:      0
Teleporter Address:   <NULL>
Teleporter Password:  <NULL>
NIC 1:               MAC: 0800270213D4, Attachment: NAT, Cable connected: on, Trace: off
(file: none), Type: 82540EM, Reported speed: 0 Mbps
NIC 2:               disabled
NIC 3:               disabled
```

*continues*



```

NIC 4:          disabled
NIC 5:          disabled
NIC 6:          disabled
NIC 7:          disabled
NIC 8:          disabled
UART 1:         disabled
UART 2:         disabled
Audio:          disabled
Clipboard Mode: Bidirectional
VRDP:           disabled
USB:            disabled
USB Device Filters: <none>
Shared folders: <none>
Guest:
Statistics update:          disabled

```

This guest should need about 512 MB of memory, and you may want to use the audio device. Also, 8 MB is not enough video RAM, and 3D acceleration is supported with the Guest Additions.

```

% VBoxManage modifyvm "Windows XP" --memory 512 --vram 32 \
    --accelerate3d on --audio oss

```

Storage configuration requires a few commands to complete. First you must create the IDE controller and the boot disk image, and attach them together. Then you must register and attach the CD-ROM image of the boot media. Finally, the BIOS boot order must be set to boot from CD-ROM before the disk. Note that because the image file name is specified as a relative path name, it is relative to the VirtualBox settings, not your current working directory.

```

% VBoxManage storagectl "Windows XP" --name "IDE Controller" \
    --add ide --controller ICH6

% VBoxManage createhd --filename "Windows XP.vdi" --size 10240 \
    --format VDI --remember

0%...10%...20%...30%...40%...50%...60%...70%...80%...90%...100%

Disk image created. UUID: 388311ed-b303-4405-8689-3dee5abc8f68

% VBoxManage storageattach "Windows XP" \
    --storagectl "IDE Controller" --port 0 --device 0 \
    --type hdd --medium «Windows XP.vdi»

```

```
% VBoxManage openmedium dvd /export/iso/windows/winxp_sp3_x86.iso

% VBoxManage storageattach "Windows XP" \
    --storagectl "IDE Controller" --port 1 --device 0 \
    --medium "/export/iso/windows/winxp_sp3_x86.iso" \
    --type dvddrive

% VBoxManage modifyvm "Windows XP" --boot1 dvd --boot2 disk
```

At this point, the virtual machine is configured and ready to boot.

### 5.5.4 Starting a Virtual Machine

Oracle VM VirtualBox provides three different methods for starting a virtual machine. The most common method is from inside the VirtualBox graphical tool itself. Select the guest and click the Start button. When the guest machine starts, a fully featured console window is opened on the host. Until the Guest Additions are installed, keyboard and mouse operations are captive. In other words, once you click inside the guest window, it will restrict your cursor to its window and will continue to receive all keyboard input until told to release the focus, usually achieved by clicking the Host key. The current Host key is displayed at the lower-right corner of the guest console window. Once the Guest Additions are installed, however, the keyboard and mouse will be seamlessly attached and detached as your mouse moves over the window.

Using this method, the user on the host can control many of the operations of the guest, such as changing the size of the window, attaching CD-ROM media, or turning seamless mode on or off.

If a less complicated console is desired, a simple window with no menu decorations can be used to start a guest. Give the command `/usr/bin/VBoxSDL` to start the guest. Instead of a fully featured window, a simple window is displayed. This approach is primarily used for debugging purposes but can be used if a simple console is required.

A more interesting start method is a headless system. In this case, the guest machine is started by the `/usr/bin/VBoxHeadless` command. This command starts the guest machine but does not display the console on the host system. Instead, the built-in Virtual Remote Desktop Protocol (VRDP) server starts. A remote system can then access the guest console using any RDP client program. For Oracle Solaris, one such program is `rdesktop`, which can be found on the Solaris Companion Software CD.

The following example demonstrates starting a guest machine in headless mode on a host system. This is just the type of operation that the Solaris Service Manage Facility (SMF) could easily automate.

```
% VBoxHeadless --startvm "Windows XP"
Sun VirtualBox Headless Interface 3.1.4
(C) 2008-2010 Sun Microsystems, Inc.
All rights reserved.

Listening on port 3389.
```

To connect to this guest from a remote system, we will use an RDP client. This example shows the use of `rdesktop` on an Oracle Solaris system to connect to the newly created Windows XP guest machine, which is running on a host named `pandora`.

```
% rdesktop pandora:3389
```

On the remote system, a new window is opened showing the guest desktop that is running on `pandora`. Figure 5.21 shows the guest desktop running the Windows XP installation program.

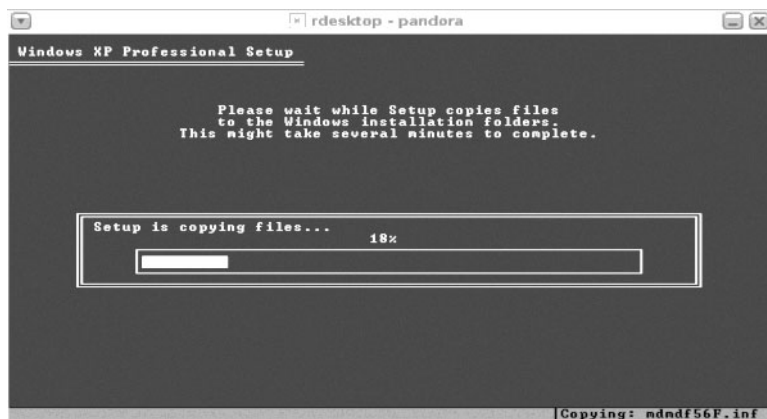


Figure 5.21 Displaying the Remote Desktop

### 5.5.5 Stopping a Virtual Machine

A guest is typically shut down by using its native method. Nevertheless, other options for stopping a virtual machine are available. Regardless of the method used to create the guest, it can be stopped from the host either through the command

line or via the VirtualBox GUI. Both of these methods provide several mechanisms that can stop a guest.

- **An ACPI shutdown signal.** This is the recommended method. A shutdown signal is sent to the guest and, if that signal is supported, the guest will begin a normal shutdown. This process includes tasks such as flushing disk caches and unmounting file systems. An ACPI signal is the safest of the external shutdown methods.
- **Snapshot.** The guest machine's current state will be saved and can be restarted from this point in the future. The snapshot approach is similar to a hibernation feature found on modern laptops, but you can save many snapshots and roll them back as desired. This flexibility enables you to test the effects of various operations, starting with a consistent state each time.
- **Reset.** This “hard stop” will cause a machine reset; it is recommended only when the ACPI shutdown does not work. In this case, the guest disk buffers are not flushed, and some of the uncommitted disk data may be corrupted. Most modern operating systems feature some form of recovery in the event of a power loss, so the chances of corruption are small, but this approach is recommended only when no other method works.
- **Pause.** This option causes a virtual machine to stop execution. The guest remains in memory, however, and its operation can be resumed. The state is not saved by a pause operation.

Here is an example of a safe external host-initiated shutdown for the guest named Windows XP:

```
% VBoxManage controlvm "Windows XP" acpipowerbutton
```

This command sends an ACPI shutdown signal to the guest, which will start an orderly shutdown process.

### 5.5.6 Cloning a Virtual Machine

Although VirtualBox does not currently feature complete cloning of a virtual machine, it does provide a host-agnostic method of cloning disk images. This is done via the `VBoxManage clonehd` command. The source disk image is copied in its entirety and a new UUID is assigned. The disk can be registered in the user's Media Manager or copied to another system to be used elsewhere. All that is required to complete the cloning of the virtual machine is to create the new guest configuration, using either the command line or the GUI as shown before.

If the host system is running Oracle Solaris, ZFS data set cloning can be used instead of cloning the virtual disk device with `VBoxManage`. The result is much faster cloning time. Because ZFS needs to allocate space only for the additional blocks that are changed in the clone, this approach also results in a smaller total storage requirement for the new guest.

To use ZFS cloning, two steps must be completed. The first step is the creation of the ZFS clone, which requires two commands. Because ZFS cloning occurs on a data set level, you should place the guest disk images in separate ZFS file systems. If a guest machine has several disks and they will always be used together, place them all in the same ZFS file system so that one ZFS operation can manage all of them.

The second step is the creation of a new UUID for the cloned virtual disk. Without this step, the cloned disk image would have the same UUID as a disk that is already registered. VirtualBox provides a method of creating a new UUID that must be used after completing the ZFS cloning operation.

The following example compares the time and space required to clone a disk image. The source is a 5 GB Solaris 10 boot disk. To simplify management of virtual disk devices, the source disk image is placed in a ZFS file system named `pandora/HardDisks/s10guest1`, which is mounted on the host as `/vbox/HardDisks/s10guest1`.

We first clone the disk image into the directory `/vbox/HardDisks/s10guest2-trad` using the VirtualBox `clonehd` method. For a consistent comparison of used space, the target directory is also a ZFS file system.

```
# time VBoxManage clonehd \
    "/vbox/HardDisks/s10guest1/Solaris 10 boot disk.vdi" \
    "/vbox/HardDisks/s10guest2-trad/Solaris 10 boot disk.vdi" \
    --remember

0%...10%...20%...30%...40%...50%...60%...70%...80%...90%...100%

Clone hard disk created in format 'VDI'. UUID: 19fb45a2-3773-4580-9e85-bb54af784d9a

real    9m40.680s
user    0m2.233s
sys     0m2.352s
```

Now we can clone the disk image using ZFS. The specific steps to do so are shown below. To record the total time of this operation, these commands were run from a script. The resulting time is indicated after the commands.

```
# zfs snapshot pandora/vbox/HardDisks/s10guest1@s10guest2-zfs

# zfs clone pandora/vbox/HardDisks/s10guest1@s10guest2-zfs \
    pandora/vbox/HardDisks/s10guest2-zfs

# VBoxManage internalcommands sethduid \
    "/vbox/HardDisks/s10guest2-zfs/Solaris 10 boot disk.vdi"
UUID changed to: f06b5f4e-805e-4e74-8345-de9a22e39e81

# VBoxManage openmedium disk \
    "/vbox/HardDisks/s10guest2-zfs/Solaris 10 boot disk.vdi"

real    0m1.288s
user    0m0.087s
sys     0m0.132s
```

Using VBoxManage, virtual disk cloning took nearly 10 minutes, and the ZFS method just slightly more than 1 second. Although all directories indicate that they are 5 GB in size, the ZFS space listing shows how much space is *actually* consumed.

```
# du -h /vbox/HardDisks
5.0G    /vbox/HardDisks/s10guest2-zfs
5.0G    /vbox/HardDisks/s10guest1
5.0G    /vbox/HardDisks/s10guest2-trad
15 G    /vbox/HardDisks

# zfs list -r -o space pandora/vbox/HardDisks
```

NAME	AVAIL	USED	USED SNAP
pandora/vbox/HardDisks	57.0G	10.0G	0
pandora/vbox/HardDisks/s10guest1	57.0G	5.00G	0
pandora/vbox/HardDisks/s10guest2-trad	57.0G	4.99G	0
pandora/vbox/HardDisks/s10guest2-zfs	57.0G	164K	0

The traditional method of using the VirtualBox copy method consumed an additional 5 GB of disk space, which was expected. However, the ZFS clone consumed only 164KB of space. Of course, as the cloned guest machine starts writing to the disk, additional space will be required for storage of its data. For the parts of the disk that are not written, only one copy of the data will be kept across all of the cloned images, which represents a significant storage savings opportunity achieved by using ZFS for guest disk images.

### 5.5.7 Live Migration of a Guest

Beginning with version 3.1, VirtualBox includes a feature called *teleportation* that allows a guest machine to move from one host to another while the guest machine is running. Except for some rare cases, the source and destination hosts need not run the same operating system or even the same type of hardware. In other words, you can migrate a guest running on an Oracle Solaris host to a Linux or Windows host.

The following steps are required to migrate a guest machine:

1. Ensure there is a TCP/IP network connection between the source and target hosts. The migration will occur over a TCP connection.
2. Configure the original guest to use some sort of shared storage (NFS, SMB, CIFS, or iSCSI) for all of its disk, CD-ROM, and floppy images.
3. On the target system, create a guest configuration that exactly matches the hardware settings (e.g., processor, memory, network) of the guest that is currently running on the source host.
4. On the target host, the guest machine must start listening for a teleportation connection request instead of actually starting. The `VBoxManage modifyvm --teleporter` command will perform this task.
5. Start the guest machine on the target host. Instead of starting, it will display a progress bar while waiting for the teleportation request from the source.
6. Initiate the live migration by issuing a `VBoxManage controlvm teleport` command on the source host system.

In this example, a guest machine named `Solaris10` migrates from a host named `source` to one named `target`. The `Solaris10` guest configurations on both hosts meet the guidelines listed above. On the target, place `Solaris10` in teleportation mode and start it. Because it is not currently in use, port 6000 will be used for the teleportation connection.

```
target% VBoxManage modifyvm Solaris 10 --teleporter on \
--teleporter 6000
target% VBoxManage startvm Solaris10
```

On the host `source` where the guest `Solaris10` is currently running, initiate the live migration with the following command:

```
source% VBoxManage controlvm Solaris10 teleport --host source \
--port 6000
```

The state of the guest `Solaris10` will be transferred to the host target and the guest will resume execution on the new host. For more information on guest teleportation, see the *VirtualBox User Manual*.

## 5.6 Summary

---

Oracle VM VirtualBox is a compact and efficient virtualization solution for Intel and AMD x86 systems. Each guest runs in a separate virtual machine and needs no additional software or drivers to run. To improve performance and allow the guests greater access to resources within the host platform, Guest Additions are provided for all supported guest operating systems.

While not as efficient as Oracle Solaris Containers, VirtualBox can take advantage of many of the resource management facilities available in Oracle Solaris to provide an excellent and well-managed environment for hosting a wide variety of applications. While features such as the internal RDP server and seamless mode make VirtualBox an obvious choice for virtualizing desktops, its performance and use of advanced virtualization features in modern hardware also make it a good choice for handling server workloads.





# Index

---

- A**
    - ABE (Alternate Boot Environment), 322
    - Accelerators, cryptographic, 88
    - Access to shared NFS resources, 272
    - ACM (Association for Computing Machinery), 329
    - ACPI shutdown signals, 163
    - Acquisition costs savings, 3
    - Adaptive replacement cache (ARC), 146-147
    - add capped-memory command, 205
    - add device command, 215
    - addboard command, 63-65, 68
    - addfru command, 66-68
    - Address spaces, multiple, 341-343
    - Agents
      - Ops Center, 316-317
      - Oracle VM, 258
    - Agility, business, 7
    - Containers, 234
    - Logical Domains, 232
    - overview, 311-312
    - virtualization for, 229
    - x86 hypervisors, 233
  - Alternate Boot Environment (ABE), 322
  - AMD PCNet FAST III, 278
  - AMD Virtualization Technology (AMD-V)
    - CPUs, 125, 130
    - HVM support, 116
  - Apache web server, 285-290
  - Application programming interface (API)
    - inter-user messaging, 332
    - VirtualBox, 129, 132
  - Applications of virtualization, 241
    - Container consolidation, 285-290
    - Container hardening, 290-300
    - Dynamic Domain configuration, 241-242
      - with CMU sharing, 243-248
    - CPUs, 248-249
    - domain configuration, 246-248, 251-254
    - I/O, 244-246, 249-251
    - isolated domains, 248-254
    - memory, 243, 249
    - summary, 254
  - Logical Domains, 254-257
  - Microsoft Windows in Containers, 276-285
  - Oracle VM Server for x86, 258-268
  - summary, 300
  - xVM hypervisor live migration, 268-276
- ARC (adaptive replacement cache), 146-147
  - Assets, 303
    - discovering, 318-320
    - managing, 325-326
    - monitoring, 323-325
    - provisioning, 320-322
    - updating, 322-323
  - Association for Computing Machinery (ACM), 329
  - Asynchronous workloads, 18
  - Atlas Ferranti, 338
  - attach command, 218
  - Auditing
    - Ops Center, 323
    - VE life cycle, 304-306
  - autoboot property, 95, 182
  - Autostart service, 282-283
  - Availability
    - domains, 57-59
    - workload consolidation, 16-18
- B**
    - Balancing
      - load, 306, 317-318, 321
      - memory, 45
    - Balloon down events, 271
    - Balloon (memory) inflation, 341

- Balloon processes, 270-271
- Bandwidth in workload
  - consolidation, 11-12, 16
- Belady, Lajos, 338
- Binding resources to Logical Domains, 89
- BIOS for VirtualBox, 143
- Boot disks for guests, 153, 156
- Boot environments
  - ABE, 322
  - xVM hypervisor, 270-271
- bootargs property, 182
- Booting Containers, 186-188
- Bound state, 105
- Boundaries for Containers, 174-177, 215, 291
- brand property, 182
- Brands for Containers, 177-178, 182
- Bridge menu, 262
- Bridged mode in VirtualBox, 142
- Browser User Interface (BUI), 60, 315-316
- Brunette, Glenn, 300
- Bursty workloads, 18-19
- Business agility, 7
  - Containers, 234
  - Logical Domains, 232
  - overview, 311-312
  - virtualization for, 229
  - x86 hypervisors, 233
- Business continuity, 312-313
- BusLogic SCSI controllers, 138
- C**
- Cache misses, 79
- Cache sharing, false, 84
- Caches, CPUs, 79-80
- Cambridge Monitor System (CMS), 330
- Capacity planning in VE life cycle, 304
- CAPP (Controlled Access Protection Profile), 175, 221
- Center for Internet Security, 300
- Centralized control, 38
- cfgadm command, 72-73
- Chip multithreading (CMT)
  - servers, 34
  - cores and processes, 84-85
  - cryptographic accelerators, 88
  - firmware updates, 322
  - Logical Domains, 77, 79-80, 82
  - installing, 89-92
  - migrating, 104
  - P2V conversion, 106
  - provisioning, 321
  - paravirtualization, 335
  - threads, 84-85, 231
  - virtual pools, 317
- Cipher/hash units, 88
- Citrix XenServer platform, 34-35, 120-121
  - drivers, 120-121
  - features, 120
  - hypervisor, 34
  - support, 121
- Clipboard for VirtualBox, 130, 144
- Clock skew, 335-336
- clone command, 216-217
- Cloning
  - Containers, 216-217, 283-285
  - guests, 163-165
  - Logical Domains, 103-104
  - VirtualBox, 146
- Clusters, 16-17
- CMS (Cambridge Monitor System), 330
- CMT. *See* Chip multithreading (CMT) servers
- CMUs (CPU/memory units), 46
  - domains, 62-63
  - sharing, 243-248
- Cold migration, 21, 105
- Command-line interface (CLI)
  - domain management, 59-60
  - Enterprise Controller, 315
  - guests, 158-161
  - Microsoft Hyper-V Server, 124
  - partitions, 44
  - VirtualBox, 133
- Common Criteria Certification, 175, 221-222
- Compatibility
  - Containers, 234
  - Dynamic Domains, 230
  - HCL, 114
  - Logical Domains, 78, 231
  - Ops Center, 319
  - software, 228
  - x86 hypervisors, 233
- Compatible Time-Sharing System (CTSS), 328-329
- Complexity of virtualization, 6
- Compute efficiency of Containers, 225
- Connected mode in Ops Center, 318
- Connectivity for Logical Domains, 86, 93-94
- Console operating system (COS), 121
- Consoles for Logical Domains, 88
- Consolidation, workload, 5-7, 9-12
  - availability, 16-18
  - Containers for, 285-290
  - performance effects, 305
  - resource controls, 12-16
  - virtualization technologies for, 237-239
- Constraints, scalability, 24-25
- Containers. *See* Oracle Solaris Containers
- Context switches, 79
- Continuity, business, 312-313
- Control domains, 34
  - configuring, 92-93
  - observing, 98-100
  - overview, 80-81
  - purpose, 321
- Controlled Access Protection Profile (CAPP), 175, 221
- Conversion, physical to virtual (P2V), 22-23, 106-108, 219, 255
- Corbató, Fernando, 328
- COS (console operating system), 121
- Costs
  - increases, 7
  - savings, 3, 6
- Cowles, Robert, 338
- CP-40 system, 329-330
- CPU caps, 197-199
- CPU IDs, 71
- CPU/memory units (CMUs), 46
  - domains, 62-63
  - sharing, 243-248
- CPU resource managers, nested, 337-338
- CPUs
  - asynchronous workloads, 18
  - Containers, 172, 180, 190-202, 211
  - domains, 49, 61
  - Dynamic Domains, 70-72, 242-243, 248-249
  - flexibility, 309-310
  - guests, 261, 266

- hardware partitions, 29-31
  - IP web service hosting, 238
  - Logical Domains, 78-80, 84-85, 102-103, 231
  - M-Series servers, 48
  - OSV efficiency, 41
  - utilization issues, 3-5
  - VirtualBox, 134-136
    - as virtualization selection factor, 236
    - workload consolidation, 10-13
  - XSBs, 50-51
  - Cray CS6400 server, 31
  - Crossbar board units (XBUs), 46-47
  - Crossbow, 222
  - Cryptographic accelerators, 88
  - CS6400 servers, 31
  - CTSS (Compatible Time-Sharing System), 328-329
- D**
- Dashboards
    - Containers, 210-211
    - Ops Center, 323
  - Data center management (DCM)
    - tools, 303-306
  - Data centers with Ops Center, 318
  - Database server management, 314
  - DCM (data center management)
    - tools, 303-306
  - dedicated-cpu feature, 201-202, 211
  - Default brands, 178
  - Defense in depth strategy, 291
  - Denning, Peter, 338
  - detach command, 217
  - /dev directory, 215
  - Device access for Containers, 215
  - DHCP servers, 86, 141, 263
  - DIAGNOSE instruction, 336
  - Directories for Containers, 178-180
  - Disaster recovery, 313-315
  - Disconnected mode in Ops Center, 318
  - Discover stage in VE life cycle, 303
  - Discovery of assets, 318-320
  - Disk images
    - golden master, 312
    - guests, 154-155
    - VirtualBox, 138-139
  - Disks
    - Containers, 172
    - guests, 153-154, 261
    - Logical Domains, 86-88
    - VirtualBox, 137-140
    - workload consolidation, 10
  - DISM (Dynamic ISM), 207
  - dispadm command, 191, 287
  - Diverse OS consolidation, 239
  - dladm create-etherstub
    - command, 223
  - dladm show-link command, 272
  - DNS proxy for VirtualBox, 141
  - Dom0 domains, 115-117
  - Dom0 guests, 34
  - Domains, 2
    - dynamic. *See* Dynamic Domains
    - logical. *See* Logical Domains
  - Domainstop operations, 73
  - DomU domains, 115
  - Double paging, 339
  - DR (Dynamic Reconfiguration)
    - Dynamic Domains, 44, 73-75
    - Logical Domains, 82, 101
  - Drivers
    - Citrix XenServer, 120-121
    - Microsoft Windows Server 2008 Hyper-V, 124-125
    - Oracle VM Server for x86, 117
    - Red Hat Enterprise Virtualization Hypervisor, 127
    - VMware ESX, 122-123
    - xVM hypervisor, 119
  - DTrace Toolkit, 207, 211-213
  - dumpadm.conf error, 284
  - Dynamic Domains, 43
    - availability, 57-59
    - building examples, 60-69
    - combinations, 54-57
    - configurations, 48-54
    - configuring, 241-242
      - with CMU sharing, 243-248
    - CPUs, 248-249
    - domain configuration, 246-248, 251-254
    - I/O, 244-246, 249-251
    - isolated domains, 248-254
    - memory, 243, 249
    - summary, 254
  - Containers in, 235
  - Dynamic Reconfiguration, 73-75
    - extending, 75-76
    - fault isolation, 73, 230
    - hard partitions, 31, 44, 230-231
    - implementing, 44-48
    - managing, 59-60
    - viewing, 69-73
  - Dynamic ISM (DISM), 207
  - Dynamic Reconfiguration (DR)
    - Dynamic Domains, 44, 73-75
    - Logical Domains, 82, 101
  - Dynamic Resource Pools, 192-199
  - Dynamic resources
    - consumption, 307-308
    - management, 102-103
  - Dynamic workload problems, 308-309
  - Dynamically expanding disk images, 138
- E**
- EAL (Evaluation Assurance Level), 221
  - EC (Enterprise Controller), 315-316
  - Efficiency, 3
    - Containers, 225, 234
    - Logical Domains, 232
    - OSV, 41
    - x86 hypervisors, 233
  - EFI (Extended Firmware Interface), 143
  - Electricity costs savings, 3
  - Elliptical curve cryptography, 88
  - Encryption in migration, 105
  - Enterprise 10000 hardware partitions, 31
  - Enterprise Controller (EC), 315-316
  - EPT (Extended Page Table), 122
  - Error-checking codes, 88
  - ESX product, 33-35, 121-122
    - drivers, 122-123
    - features, 122
    - support, 123
  - ESXi product, 121
  - /etc/apache2/httpd.conf file, 285
  - /etc/dhcp files, 94

- /etc/hostname files, 94
- /etc/httpd/conf/httpd.conf file, 285-286
- /etc/inet/hosts file, 275
- /etc/init.d/vbox file, 282
- /etc/ldmp2v.conf file, 106
- /etc/nsswitch.conf file, 177
- /etc/patch/pdo.conf file, 180
- /etc/sysidcfg file, 188, 216-217, 279, 289
- /etc/vfstab file, 289-290
- Ethernet communication, 60
- Evaluation Assurance Level (EAL), 221
- Exclusive-IP Containers, 213-215
- export command, 216
- Extended Firmware Interface (EFI), 143
- Extended Page Table (EPT), 122
- Extended System Control Facility (XSCF), 44, 73-74
- Extending Dynamic Domains, 75-76
- F**
- Failure isolation
  - containers, 173
  - hardware partitions, 28-29
  - OSV, 37
  - virtual machines, 32
- Fair Share Scheduler (FSS), 337
  - Containers, 190-192, 198
  - as default scheduler, 287
  - workload consolidation, 12-14
- False cache sharing, 84
- Fast cloning in VirtualBox, 146
- Fault isolation
  - Containers, 233
  - Dynamic Domains, 73, 230
  - Logical Domains, 231
  - x86 hypervisors, 232
- Fault Management system, 75
- File systems for Containers, 178-180, 183-185
- Fine-grained operating system modification, 25
- Firmware
  - Logical Domains, 90
  - provisioning, 320
- Fixed-size image files, 138
- flarcreate command, 106
- Flash Archive (FLAR), 321
- Flexibility
  - operational, 229, 309-311
  - resource configuration, 29-30, 39-40
  - system virtualization models, 26
- Flexible provisioning, 23-24
- Folders for VirtualBox, 145
- FSS (Fair Share Scheduler), 337
  - Containers, 190-192, 198
  - as default scheduler, 287
  - workload consolidation, 12-14
- Full virtualization, 35
- Future workloads, planning for, 8
- G**
- General Public License (GPL), 115
- Global control and observability, 38
- Global properties for Containers, 182
- Global zones
  - Containers, 171-172, 175, 278
  - VirtualBox, 145
- GNU General Public License (GPL), 115
- Goldberg, Robert P., 329-330
- Golden master images, 312
- Granularity
  - Containers, 234
  - Logical Domains, 232
  - resource configuration, 29-30, 38-40
  - vCPU assignments, 84
  - x86 hypervisors, 233
- Graphical User Interface (GUI)
  - domain management, 60
  - Logical Domains, 108
  - VirtualBox, 133, 163
- Graphics in VirtualBox, 144
- Guest Additions, 144-145, 150
- Guests, 33-35
  - autostart service, 282-283
  - creating, 259-260
  - information for, 260-262
  - installation media, 260
  - instruction simulation, 333
  - interactive installation, 263
  - live migration, 166-167, 273-274
  - Logical Domains, 81
  - Microsoft Windows in Containers, 280-283
  - network connectivity, 93-94
  - network interface reconfiguration, 264-267
- Oracle Solaris 10
  - deployment, 259-267
  - provisioning, 262-263
  - scalability, 30
  - synthetic instructions for, 336-337
- VirtualBox. *See* Oracle VM VirtualBox
- virtualization method, 260
- x86. *See* x86 guests
- GUI (Graphical User Interface)
  - domain management, 60
  - Logical Domains, 108
  - VirtualBox, 133
- H**
- HA (high-availability) solutions, 16
- Halting Containers, 188-189
- Handshaking mechanism, 340
- Hardening security for Containers. *See* Oracle Solaris Containers
- Hardware
  - Container independence, 225
  - isolation, 227-228
  - OSV, 36, 39
- Hardware Compatibility List (HCL), 114
- Hardware partitions, 28, 44
  - Dynamic Domains, 31, 44, 230-231
  - failure isolation, 28-29
  - industry examples, 31
  - managing, 30
  - operating systems, 29
  - relative strengths, 31
  - resource configuration, 29-30
  - scalability, 30
- Hardware virtual machines (HVMs), 115-116
- Harvester tool, 318
- HCL (Hardware Compatibility List), 114
- Headless systems, 161
- Heat generation savings, 2
- Hewlett-Packard nPars feature, 31
- High-availability (HA) solutions, 16
- History of system virtualization, 327

- clock skew, 335–336
- hypervisors, 328–330
- memory management, 338–343
- nested CPU resource managers, 337–338
- performance challenges, 333–335
- synthetic instructions, 336–337
- virtual machines, 330–332
- Holistic observability, 211–213
- Host key, 161
- Host only mode in VirtualBox, 142
- Hosting IP web services, 238–239
- Hot plugs, 75
- httpd status command, 286
- httpd stop command, 286
- HVM+PVIO guests, 273–274
- HVMs (hardware virtual machines), 115–116
- Hybrid I/O in Logical Domains, 86
- Hypervisors, 2
  - clock skew, 335–336
  - history, 328–330
  - Hyper-V. *See* Hyper-V
  - KVM. *See* KVM
  - live migration. *See* Live migration
  - Logical Domains. *See* Logical Domains
  - Oracle VM. *See* Oracle VM
  - relative strengths, 36
  - system virtualization models, 27
  - Type 1, 32–34, 114–115
  - Type 2, 34–35
  - VirtualBox, 129, 132–133. *See also* VirtualBox
  - VMware. *See* VMware
  - workload consolidation, 15
  - x86 guests, 114–116, 118–119
- I**
  - IBM hypervisors, 33–34
  - ifconfig command, 85, 93–94
  - ILOM processor, 323
  - Image file formats, 138–139
  - Immutable Service Containers (ISCs), 177
  - Inactive state for migration, 105
  - info command, 182
  - inherit-pkg-dir property, 183
  - init process, 199–200
  - Instances
    - Containers, 217
    - Logical Domains, 232
    - x86 hypervisors, 233
  - Instruction simulation performance effects, 333–335
  - Intel VT CPUs, 116, 125, 130
  - Interactive guest installation, 263
  - Interactive Text (Console Session) option, 263
  - Internal mode in VirtualBox, 142
  - Interprocess communication (IPC) mechanisms, 37
  - Interrupts, 334
  - Intimate Shared Memory (ISM), 207
  - I/O
    - domains, 34, 49, 62, 81
    - Dynamic Domains, 244–246, 249–251
    - Logical Domains, 82–83, 86
    - SPARC Enterprise M-Series servers, 48
    - system controller transactions, 45
    - workload consolidation, 10–11, 14
  - ioemu interface, 261, 266
  - IOMMUs (I/O memory mapping units), 82
  - IOUs (I/O units), 45–47
    - domains, 62–63
    - Dynamic Domains, 244–245, 249–251
  - IP addresses
    - cloned instances, 104–105
    - Containers, 218, 277–278, 286, 291, 293
    - guests, 262
    - host name mapping, 177
    - port redirection, 141
    - VEs, 314
    - virtual network devices, 85
  - IP Filter, 290, 296, 300
  - IP Multipathing (IPMP), 83, 85, 214
  - ip-type property, 182
  - IP web service hosting, 238–239
  - IPC (interprocess communication) mechanisms, 37
  - ipf command, 225
  - ipkg brand, 178
  - IPMP (IP Multipathing), 83, 85, 214
  - ipnat command, 225
  - ISCs (Immutable Service Containers), 177
  - ISM (Intimate Shared Memory), 207
  - Isolated data centers, 318
  - Isolation, 227–228
    - Containers, 173–177, 233
    - domains, 45
    - Dynamic Domains, 73, 230
    - hardware partitions, 28–29
    - Logical Domains, 78, 231
    - OSV, 37
    - Red Hat Enterprise Virtualization Hypervisor, 126
    - system virtualization models, 26
    - virtual machines, 32
    - x86 hypervisors, 232
  - J**
    - Jails technology, 2, 169
    - Java-based VNC client, 263
    - JumpStart Enterprise Toolkit (JET), 320
  - K**
    - KB (knowledge base), 316
    - Kernel-based Virtual Machine (KVM) technology, 125–126
    - kernel data structures for workload consolidation, 12
    - Keyboards in VirtualBox, 144
    - Knowledge base (KB), 316
    - kstat command
      - Containers, 190, 198
      - NICs, 214
      - virtual memory, 204–205, 209
    - KVM (Kernel-based Virtual Machine) technology, 125–126
  - L**
    - Labeled Security Protection Profile (LSPP), 175, 221
    - Large workloads, virtualization technologies for, 237–238

- Latency
    - memory access, 53-54
    - workload consolidation, 10, 14
  - Latency-sensitive workloads, 229
  - LDCs (Logical Domain Channels), 83
  - ldm command for control
    - domains, 81, 91-93
  - ldm add-vcpu command, 101
  - ldm add-vdisk command, 87, 101
  - ldm add-vdsdev command, 87
  - ldm add-vnet command, 101
  - ldm bind command, 89, 96
  - ldm list command, 84, 98
  - ldm migrate command, 104-105
  - ldm set-vcpu command, 101
  - ldm start command, 96
  - ldmconfig command, 108
  - ldmp2v command, 106-108, 256
  - Least recently used (LRU)
    - pages, 339-340
  - Legacy operating systems, 23
  - /lib directory, 178-180
  - Libraries
    - Ops Center, 317
    - OS profiles, 321
  - Licenses
    - Containers, 234
    - CPUs, 199
    - Dynamic Domains, 230
    - GNU, 115
    - Logical Domains, 232
  - Life cycles
    - Container, 189
    - VEs, 303-306
  - Lightweight processes (LWPs), 209-210
  - limitpriv property, 182
  - Live migration, 21-22, 105, 268
    - configuring and enabling, 274-276
    - guests, 166-167
    - HVM+PVIO guests, 273-274
    - prerequisites, 269
    - shared NFS resource access, 272
    - summary, 276
    - technical briefs, 268-269
    - xVM hypervisor configuration, 269-272
  - Live Upgrade feature, 322
  - Load-balanced cluster pairs, 16-17
  - Load balancing, 306, 317-318, 321
  - Locked memory, 15, 207-209
  - Logical Domain Channels (LDCs), 83
  - Logical Domain Manager
    - services, 81
  - Logical Domains, 77, 254-255
    - advantages, 231-232
    - cloning, 103-104
    - configuring, 256
    - consoles, 88
    - vs. Containers, 109
    - Containers in, 235
    - control domains, 92-93, 98-100
    - creating, 94-95
    - cryptographic accelerators, 88
    - domains creation, 256-257
    - dynamic reconfiguration, 82, 101
    - dynamic resource management, 102-103
    - ease-of-use enhancements, 108-109
    - features, 77-80
    - firmware, 90
    - memory, 83, 89
    - mobility, 104-105
    - network connectivity, 86, 93-94
    - Ops Center data, 325
    - physical to virtual
      - conversion, 106-108
    - planning, 255-256
    - provisioning, 321
    - relationships, 81-82
    - resource binding, 89
    - roles, 80-82
    - software, 90-92
    - Solaris installed into, 97-98
    - summary, 110-111, 257
    - testing, 257
    - viewing, 96-97, 100-101
    - virtual CPUs, 84-85
    - virtual disks, 86-88
    - virtual I/O, 82-83
    - virtual network devices, 85-86
  - Logical Domains Configuration Assistant, 81, 108-109
  - Logical Domains Dynamic Resource Manager, 84
  - Logical Domains manager, 86
  - Logical Partitions (LPARs), 31, 34
  - Logical system boards (LSBs), 70-72
  - Loopback mounts for
    - Containers, 184
  - Low-level I/O transactions, auditing, 305
  - LPARs (Logical Partitions), 31, 34
  - LRU (least recently used) pages, 339-340
  - LSBs (logical system boards), 70-72
  - LSI Logic controllers, 138
  - LSPP (Labeled Security Protection Profile), 175, 221
  - LWPs (lightweight processes), 209-210
- M**
- M4000/M5000 systems
    - block diagram, 45-46
    - characteristics, 48
    - domain combinations, 54-55
    - Dynamic Reconfiguration, 75
    - Quad-XSB, 51-54, 57-58
    - Uni-XSB, 50-51
    - block diagram, 46-47
    - characteristics, 48
    - domain availability, 57-59
    - domain combinations, 55-56
    - Dynamic Reconfiguration, 74-75
    - Quad-XSB, 51-54
    - Uni-XSB, 50-51
  - M9000 configuration, 242
    - with CMU sharing, 243-248
    - CPUs, 248-249
    - domain configuration, 246-248, 251-254
    - I/O, 244-246, 249-251
    - isolated domains, 248-254
    - memory, 243, 249
    - summary, 254
  - MAC (memory access controller), 45, 57-59
  - MAC addresses
    - Containers, 291
    - guests, 262
    - Logical Domains, 85-86, 100, 106
    - VirtualBox, 142
  - Manage stage in VE life cycle, 306

- Massachusetts Institute of Technology (MIT), 328
- Master devices with IDE controllers, 137
- MAUs (Modular Arithmetic Units), 88
- MDF (Multiple Domain Facility), 31
- Memory
  - access latency, 54
  - Containers, 172, 179, 203-209, 211
  - domains, 49, 61-62
  - Dynamic Domains, 243, 249
  - guests, 152-153, 160, 261, 266
  - IP web service hosting, 238-239
  - Logical Domains, 83, 89
  - M-Series servers, 48
  - mirroring, 57-58
  - multiple address spaces, 341-343
  - OSV, 41
  - partitioning, 15
  - reclaiming, 271
  - thrashing, 338-341
  - VirtualBox, 136-137
  - workload consolidation, 10-11, 14-15
- Memory access controller (MAC), 45, 57-59
- Memory caps, 15, 203, 205-206, 208-209
- Memory mapping units (MMUs), 83
- Microsoft Windows, updating, 323
- Microsoft Windows in Containers using VirtualBox, 276-277
- Container cloning, 283-285
- Container creation, 278-280
- global zones, 278
- guests, 280-283
- planning, 277-278
- summary, 285
- Microsoft Windows Server 2008 Hyper-V, 123-124
- drivers, 124-125
- support, 125
- Migration, 7
  - Containers, 217-219
  - guests, 166-167
  - live. *See* Live migration
  - overview, 104-105
  - types, 21-22
  - workload restoration, 314
- Minidisks, 331
- Mirroring memory, 57-58
- MIT (Massachusetts Institute of Technology), 328
- mkfile command, 95
- MLS (multi-level security) protection, 175, 221
- MMUs (memory mapping units), 83
- Mobility
  - Logical Domains, 104-105
  - workloads, 20-23
- Models, system virtualization
  - hardware partitions, 27-31
  - operating system virtualization, 36-41
  - overview, 26-28
  - virtual machines, 32-36
- Modes
  - Ops Center, 318
  - VirtualBox, 141-142
- Modular Arithmetic Units (MAUs), 88
- Modular design for VirtualBox, 130
- Monitor and audit usage stage in VE life cycle, 304-306
- Monitoring
  - assets, 323-325
  - Containers, 189-213
  - CPU usage, 199-202
  - virtual memory, 204-209
- Most recently used (MRU) pages, 340
- Motherboards in VirtualBox, 137
- mount command, 289
- Mouse for VirtualBox, 144
- mpstat command, 84, 190, 199, 201-202
- MPXIO (multiplexed I/O), 87
- MRU (most recently used) pages, 340
- Multi-level security (MLS) protection, 175, 221
- MULTICS system, 328
- Multilayered virtualization, 27
- Multiple address spaces, 341-343
- Multiple Domain Facility (MDF), 31
- Multiple processors in Oracle VM, 261
- Multiplexed I/O (MPXIO), 87
- Multiprocessing systems, 196-197
- Multithreading
  - CMT. *See* Chip multithreading (CMT) servers
  - Dynamic Resource Pools, 196-197
  - Logical Domains, 231
- Multiuser operating systems, 4
- N**
- Namespaces for Containers, 177
- NAS (network attached storage), 314-315
- NAT (Network Address Translation), 141, 225
- native brand, 178
- Nested CPU resource managers, 337-338
- Nested Page Tables (NPT), 343
- netfront interface, 261, 266
- net services command, 294-295
- Network Address Translation (NAT), 141, 225
- Network attached storage (NAS), 314-315
- Network Interface Unit (NIU) Hybrid I/O, 86
- Network Time Protocol (NTP), 291, 296, 300, 336
- Networks
  - bandwidth reservation, 16
  - bandwidth in workload consolidation, 11-12, 16
  - Containers, 213-215, 222-225
  - in disaster recovery, 314
  - interface reconfiguration for guests, 264-267
  - Logical Domains virtual devices and connectivity, 85-86, 93-94
  - VirtualBox virtual devices, 140-142
- New Virtual Machine Wizard, 151-153
- NFS shares access, 272
- NICs
  - Containers, 213-214
  - VNICs, 222-223
- NIU (Network Interface Unit) Hybrid I/O, 86
- Non-executable pages (NX), 135
- Not attached mode for VirtualBox, 141

- nPars feature, 31
  - NPT (Nested Page Tables), 343
  - NTP (Network Time Protocol), 291, 296, 300, 336
  - ntpdate command, 295, 297-298
  - NX (non-executable pages), 135
  - O**
  - OBP (OpenBoot Prom)
    - variables, 95-96
  - Observability of Containers, 211-213, 225, 234
  - Online transaction processing (OLTP), 18
  - Open Service Tags, 318
  - OpenBoot, 88, 96
  - OpenBoot Prom (OBP)
    - variables, 95-96
  - OpenSolaris
    - HCL, 114
    - networking, 222-225
    - xVM hypervisor for, 118-119
  - OpenSolaris Immutable Service Containers project, 300
  - Operating system virtualization (OSV), 2
    - auditing by, 305
    - DTrace for, 212
    - failure isolation, 37
    - hardware access, 39
    - industry examples, 41
    - operating system features, 37-39
    - overview, 36-37
    - platform management, 40
    - relative strengths, 40-41
    - resource configuration
      - flexibility and granularity, 39-40
    - scalability, 40
    - software infrastructure, 39-40
    - system virtualization models, 27
  - Operating systems (OS)
    - consolidation, 239
    - diagnosis engine, 73
    - fine-grained modification, 25
    - flexibility in, 229
    - guests, 156
    - hardware partitions, 29
    - legacy, 23
    - provisioning, 320-321
  - Operational efficiencies
    - Containers, 234
    - Logical Domains, 232
    - x86 hypervisors, 233
  - Operational flexibility, 229, 309-311
  - Opportunities for virtualization, 309-315
  - Oracle Enterprise Manager Ops Center, 60, 108-109, 315
    - assets
      - discovering, 318-320
      - managing, 325-326
      - monitoring, 323-325
      - provisioning, 320-322
      - updating, 322-323
    - concepts, 317-318
    - data centers, 318
    - Logical Domains, 81
    - structure, 315-317
    - summary, 326
  - Oracle Solaris
    - Containers. *See* Oracle Solaris Containers
    - deployment with Oracle VM Server for x86, 258
    - guests, 259-267
    - prerequisites, 258-259
    - summary, 268
    - installing into domains, 97-98
  - Oracle Solaris Cluster, 16-17
  - Oracle Solaris Containers, 2, 75, 169
    - advantages, 233-235
    - basic model, 171-173
    - booting, 186-188
    - brands, 177-178
    - capabilities, 169-170
    - cloning, 216-217, 283-285
    - consolidation, 285-286
    - Container creation, 287-290
    - planning, 286-287
    - Solaris 10 configuration, 287
    - summary, 290
    - testing, 290
  - CPU controls, 190-202
  - creating, 181-185
  - dashboard, 210-211
  - direct device access, 215
  - DTrace, 211-213
  - Dynamic Domains, 72
  - features, 170-171
  - file systems, 183-185
  - global properties, 182
  - halting, 188-189
  - hardening, 290-292
    - analysis, 298-299
    - implementing, 294-298
    - information on, 299-300
    - scenario, 292-293
    - steps, 293-294
    - summary, 299
  - installing and booting, 186-188
  - isolation, 173-177, 233
  - vs. Logical Domains, 109
  - memory controls, 203-209
  - memory management, 341
  - Microsoft Windows. *See* Microsoft Windows in Containers
  - migrating, 217-219
  - miscellaneous controls, 209-210
  - namespaces, 177
  - networking, 213-215, 222-225
  - packaging, deployment, and file systems, 178-180
  - patching, 180-181
  - privileges, 186
  - provisioning, 322
  - resource management, 189-213
  - Solaris 8 and Solaris 9, 219-221
  - strengths, 225-226
  - summary, 226
  - system virtualization models, 27
  - Trusted Extensions, 221-222
  - VirtualBox, 145-146
- Oracle VM (OVM), 258
- Oracle VM Agent, 258
- Oracle VM Manager, 258-260
- Oracle VM Server, 34
- Oracle VM Server for SPARC. *See* Logical Domains
- Oracle VM Server for x86, 116-117, 258, 260-261
  - drivers, 117
  - features, 117
  - Oracle Solaris deployment with. *See* Oracle Solaris support, 118
- Oracle VM Server Pools, 261
- Oracle VM VirtualBox, 129-130
  - architecture, 132-133



- guests, 151
    - BIOS and EFI, 143
    - boot disks, 156
    - cloning, 163-165
    - creating, 151-155, 158-161
    - Guest Additions, 144-145, 150
    - kernel code, 131
    - live migration, 166-167
    - Microsoft Windows in
      - Containers, 278, 280-282
    - New Virtual Machine Wizard, 151-153
    - operating system installation, 156-158
    - RAM, 136-137
    - Solaris as, 149-151
    - starting, 161-162
    - stopping, 162-163
    - summary, 167
    - virtual CPUs, 134-136
    - virtual disks, 137-140
    - virtual network devices, 140-142
  - installing, 147-149
  - interacting with, 133-134
  - new Virtual Machine Wizard, 151-153
  - operation, 131-132
  - Solaris support, 145-147
  - OSV. *See* Operating system virtualization (OSV)
  - Overhead, 8,
    - Containers, 225, 234
    - CPUs, 10-11, 13
    - HVM, 116
    - hypervisor, 34-36
    - instruction emulation, 333-335
    - Logical Domains, 78-79, 83
    - memory access, 341-343
    - virtual machines, 32
    - as virtualization selection factor, 228-229
- P**
- P2P, 23
  - P2V tool
    - Containers, 219
    - Logical Domains, 106-108, 255
    - migration, 22-23
  - PA-RISC-based hardware
    - partitions, 31
  - Packages
    - Containers, 178-180
    - patching, 180-181
  - PAE (Physical Address Extensions), 114-115
  - Page table entries (PTEs), 341-342
  - Paging workload consolidation, 14
  - Parallel patching of Containers, 180
  - Parallels version 2 HDD format, 138
  - Paravirtualization
    - CMT, 335
    - guests, 115-116, 266-267
    - virtual machines, 35
  - Partitions, 2
    - with Dynamic Domains, 230-231
    - hardware. *See* Hardware partitions
    - Microsoft Windows Server 2008 Hyper-V, 123-124
    - workload consolidation, 12-13, 15
  - patchadd command, 90, 180-181
  - Patching Containers, 180-181
  - Pausing guests, 163
  - PCIe cards, 72-73
  - Performance
    - Containers, 234
    - Dynamic Domains, 230
    - instruction simulation, 333-335
    - Logical Domains, 231
    - monitoring, 305
    - overhead. *See* Overhead
    - testing, 20
    - throughput, 2
    - workload consolidation, 10, 305
  - Physical Address Extensions (PAE), 114-115
  - Physical computers limitations, 307
  - Physical interrupts, 334
  - Physical System Boards (PSBs), 48-50
  - Physical to Virtual (P2V) tool
    - Containers, 219
    - Logical Domains, 106-108, 255
  - migration, 22-23
  - PIT (Programmable Interval Timer), 336
  - pkgadd command, 150
  - Planning
    - Container consolidation, 286-287
    - Logical Domains, 255-256
    - Microsoft Windows in
      - Containers, 277-278
  - /platform directory, 178
  - Platforms
    - OSV management, 40
    - VirtualBox, 129-130
  - pool property, 182
  - poolcfg command, 196, 201
  - poold service, 195
  - Pools
    - CPU, 190
    - Dynamic Resource Pools, 192-199
    - Ops Center, 317-318
    - Oracle VM Server, 261
  - poolstat command, 190, 195, 199, 201-202
  - Popek, Gerald J., 329-330
  - POST diagnosis engine, 73
  - poweron command, 63-64
  - PowerVM Hypervisor, 34
  - prctl command
    - Containers, 190-192, 197-198
    - LWPs, 209
    - virtual memory, 204, 208-209
  - Predefined workload templates, 312
  - Predictive Self-Healing
    - functions, 75
  - priocntl command, 191, 287
  - Private CPU pools, 192
  - privdebug script, 294-297
  - Privilege Debugging Tool
    - project, 300
  - Privileged domain 0, 115
  - Privileges for Containers, 174-177, 186, 293-297
  - proc\_lock\_memory privilege, 209
  - Process rights management, 174
  - Processors. *See* CPUs
  - Profiles
    - firmware, 320
    - operating systems, 321
    - templates, 312
    - VE, 304

- Programmable Interval Timer (PIT), 336
- Programmer activities, 18-19
- Project Crossbow, 222
- Provision stage in VE life cycle, 303-304
- Provisioning
  - assets, 320-322
  - flexible and rapid, 23-24
  - guests, 262-263
- Proxy Controllers, 315-316, 323-324
- prstat command
  - Containers, 190, 198-201
  - LWPs, 209
  - virtual memory, 204, 207
- prtdiag command, 69-70
- ps command, 199-200
- PSBs (Physical System Boards), 48-50
- psrinfo command, 101
- PTEs (page table entries), 341-342
- Q**
- QEMU processor emulator, 116, 131
- Quad-XSB, 49, 51-56
- R**
- RAC product, 17
- RAM. *See* Memory
- Rapid provisioning, 23-24
- Rapid Virtualization Index (RVI), 122
- RAS (reliability, availability, and serviceability)
  - characteristics, 311
- RBACPP (Role-Based Access Control Protection Profile), 175, 221
- rcapadm command, 190, 205
- rcapd daemon, 206
- rcapstat command, 190, 205-207, 298
- rdesktop program, 161-162, 281, 285
- RDP (Remote Desktop Protocol)
  - server, 129-130
- Read and write transaction
  - latency, 10
- Read-only mounts, 290
- Read-only virtual disks, 87
- README files, 90, 147
- Real CPUs in Logical Domains, 78-80
- Real Time Clock (RTC), 336
- Recertification of software, 228
- Reclaiming memory, 271
- Recovery, disaster, 313-315
- Red Hat Enterprise
  - Virtualization (RHEV)
    - products, 125-127
    - Hypervisor (RHEV-H), 125
    - Manager for Servers (RHEV-M Server), 125
- Related workloads, 308
- Relationships
  - Logical Domains, 81-82
  - workloads, 308
- Reliability, availability, and serviceability (RAS)
  - characteristics, 311
- Remote Desktop Protocol (RDP)
  - server, 129-130
- Report compliance in VE life cycle, 304
- Requirements as virtualization
  - selection factor, 235-236
- Resetting guests, 163
- Resource caps in workload
  - consolidation, 12
- Resource consumption, dynamic, 307-308
- Resource control and configuration
  - Containers, 189-213
  - flexibility and granularity, 29-30, 39-40
  - nested managers, 337-338
  - partitions. *See* Partitions
  - workload consolidation, 10, 12-16
- Restoration of VEs, 314
- RHEV (Red Hat Enterprise Virtualization)
  - products, 125-127
- RHEV-H (Red Hat Enterprise Virtualization Hypervisor), 125
- RHEV-M Server (Red Hat Enterprise Virtualization Manager for Servers), 125
- Rights management for
  - Containers, 174-177
- Rings in VirtualBox, 131-132
- Role-Based Access Control
  - Protection Profile (RBACPP), 175, 221
- Roles for Logical Domains, 80-82
- Root user for Containers, 174
- Routers
  - Containers, 224
  - VirtualBox, 141
- rpc/bind service, 295
- rpool/boot/grub/menu.lst file, 270
- RTC (Real Time Clock), 336
- rtls network driver, 261, 265
- RVI (Rapid Virtualization Index), 122
- S**
- s10x86.iso file, 273
- SANs (storage area networks), 314-315
- SATA (Serial ATA) controllers, 137-138
- Savings benefits, 2-3, 6
- /sbin directory, 178
- SC (system controller), 45
- Scalability, 8
  - constraints, 24-25
  - Containers, 234
  - hardware partitions, 30
  - OSV, 40
  - and performance, 229
- Scalable Services, 17
- SCCM (System Center Configuration Manager), 323
- scheduling-class property, 182
- SCONFIG interface, 124
- SCSI controllers, 138
- SCVMM (System Center Virtual Machine Manager), 124
- SDK (software development kit)
  - for VirtualBox, 129
- Seamless mode in VirtualBox, 130, 144
- Secure data centers, 318
- Secure hash, 88
- Security, 8
  - configurability, 25-26
  - Containers, 174-177, 179, 215, 234-235, 290-293
  - domains, 45
  - hardening. *See* Oracle Solaris Containers

- in migration, 105
- networks, 224
- Ops Center policies, 318
- Red Hat Enterprise
  - Virtualization
    - Hypervisor, 126
- Security boundaries, 174-177, 215, 291
- Security-Enhanced Linux (SELinux), 126
- Security isolation, 228
  - Dynamic Domains, 230
  - Logical Domains, 231
  - x86 hypervisors, 232
- select command for NICs, 213-214
- select capped-memory command, 205
- SELinux (Security-Enhanced Linux), 126
- Serial ATA (SATA) controllers, 137-138
- Service consoles, 33
- Service domains, 34, 80
- Service level agreements (SLAs), 12
- Service Management Facility (SMF)
  - for Apache, 290
  - Containers, 177, 186
  - in security, 291, 293
  - starting guests, 162
- Service uptime for IP web service hosting, 238
- set limitpriv command, 294
- setdcl command, 63-65, 68
- setproperty command, 149
- setupfru command, 63-65
- Shadow page tables, 342-343
- Shared Clipboard for
  - VirtualBox, 130, 144
- Shared I/O channels, 10, 14
- Shared-IP Containers, 213-214
- Shared memory
  - Containers, 207-209
  - Logical Domains, 83
- Shared NFS resources,
  - accessing, 272
- Shared storage, VEs on, 313
- showboards command, 63-65, 68
- showdcl command, 68-69
- showhardconf command, 60-62
- Simplifying workload mobility, 20-23
- SLAs (service level agreements), 12
- Slave devices with IDE controllers, 137
- Smart groups, 323
- SMF (Service Management Facility)
  - for Apache, 290
  - Containers, 177, 186
  - in security, 291, 293
  - starting guests, 162
- Snapshots
  - database, 314
  - disaster recovery, 313
  - golden masters, 312
  - guests, 163
  - VEs, 7, 20
  - ZFS, 87, 103
- SNMP support, 60
- Software
  - isolation, 227-228
  - Logical Domains, 90-92
  - OSV, 39-40
  - recertification, 228
- Software development and bursty workloads, 18-19
- Software development kit (SDK)
  - for VirtualBox, 129
- Software schedulers, 13
- Solaris. *See* Oracle Solaris
- Solaris Fault Management system, 75
- Solaris Zones. *See* Oracle Solaris Containers
- SPARC Enterprise M-Series servers, 44
  - M4000/M5000. *See* M4000/M5000 systems
  - M8000/M9000. *See* M8000/M9000 systems
  - M9000 configuration. *See* M9000 configuration
- SPARC environments as virtualization selection factors, 236-237
- Sparse-root Containers, 179-180
- ssh command, 88
- Staging, 19-20
- stmsboot command, 88
- Stopping virtual machines, 162-163
- Storage area networks (SANs), 314-315
- Storage bandwidth in workload consolidation, 12
- Storage efficiency in Containers, 225
- Storage replication tools, 313-314
- Strands, 79, 84, 99
- Sun Management Center, 60
- Sun Ray thin clients, 332
- Sun Service Tags, 318
- SUNWbox package, 147
- SUNWvbox package, 147
- SUNWvboxkern package, 147
- SVC instruction, 333
- svcadm command
  - Containers, 290, 294
  - networks, 224
- sVirt project, 126
- Swap caps, 203
- Swap reservations, 15
- Swap space used in workload consolidation, 11
- Synchronization
  - clock skew, 335-336
  - Containers, 295
  - Guest Additions, 144
  - patches, 232
- Synthetic instructions, 336-337
- sys-unconfig command, 216, 265-266
- System/360 computers, 329-331
- System/370 computers, 331
- System Center Configuration Manager (SCCM), 323
- System Center Virtual Machine Manager (SCVMM), 124
- System controller (SC), 45
- System features for isolation, 228
- System virtualization models
  - hardware partitions, 27-31
  - operating system
    - virtualization, 36-41
  - overview, 26-28
  - virtual machines, 32-36
- T**
- Teleportation, 166
- telnet command, 88, 96
- Templates, workload, 312
- Temporary CPU pools, 192
- Testing, 19-20
  - Container consolidation, 290
  - Logical Domains, 257
- Thin clients, 332

- Thrashing, 338–341
- Threads, 84–85, 231. *See also* Multithreading
- Throughput performance, 2
- Time sharing, 332
- Time-slicing, 78
- Time synchronization
  - clock skew, 335–336
  - Containers, 295
  - Guest Additions, 144
  - patches, 232
- Transactions
  - auditing, 305
  - latency, 10
  - OLTP, 18
- Translation look-aside buffers (TLBs), 79, 341–343
- Trojan Horse attacks, 179, 291
- Trusted Extensions feature, 170, 175, 221–222
- Type 1 hypervisors, 32–34, 114–115
- Type 2 hypervisors, 34–35
- U**
- UFS file systems, 86–87
- ufsdump command, 106
- UIDs (user identification numbers), 177
- UltraSPARC systems
  - cryptographic accelerators, 88
  - Logical Domains, 79–80
  - virtual CPUs, 84–85
  - virtual network devices, 86
- Uni-XSB, 49–51
- Unprivileged guest domains, 115
- Update and report compliance
  - stage in VE life cycle, 304
- Updating assets, 322–323
- Use cases, 9
  - asynchronous workloads, 18
  - fine-grained operating system modification, 25
  - flexible, rapid provisioning, 23–24
  - legacy operating systems, 23
  - scalability constraints, 24–25
  - security configuration, 25–26
  - software development and
    - bursty workloads, 18–19
  - testing and staging, 19–20
  - workload consolidation, 9–18
  - workload mobility simplification, 20–23
- User identification numbers (UIDs), 177
- User rights management for Containers, 174
- /usr directory, 178–180
- UUIDs for cloned disks, 164
- V**
- V2P, 23
- V2V, 23
- /var/apache2/htdocs directory, 286
- /var/www/html directory, 286
- VBoxGuestAdditions.iso file, 140, 144, 156, 282
- VBoxHeadless command, 148, 161
- VBoxManage command, 133, 148, 153
- VBoxManage clonehd command, 163–164
- VBoxManage controlvm
  - command, 163
- VBoxManage controlvm teleport
  - command, 166
- VBoxManage createhd
  - command, 138, 160
- VBoxManage createvm
  - command, 159
- VBoxManage list ostypes
  - command, 159
- VBoxManage modifyvm
  - command, 135, 143, 160–161, 166
- VBoxManage openmedium
  - command, 138, 161
- VBoxManage setextradata
  - command, 141
- VBoxManage sharefolder
  - command, 150
- VBoxManage showhdinfo
  - command, 139
- VBoxManage showvminfo
  - command, 135, 159
- VBoxManage storageattach
  - command, 160–161
- VBoxManage storagectl
  - command, 160
- VBoxSDL command, 148, 161
- VBoxSolarisAdditions.pkg
  - package, 150
- vboxsrv driver, 132
- VBoxSVC process, 133
- VBoxXPCOMIPCD process, 133
- vboxzoneaccess daemon, 133
- VC (Virtualization Controller), 317
- Vcc (virtual console concentrators), 88
- vCPU IDs, 71
- vCPUs (virtual CPUs), 71
  - Logical Domains, 84–85
  - VirtualBox, 134–136
- VDI (Virtual Desktop Integration), 332
- VDI disk format, 138
- Verifying Logical Domain firmware, 90
- Veritas Solaris Cluster, 16
- VEs. *See* Virtual environments (VEs)
- VHD disk format, 138
- VI (VMware Infrastructure), 121
- Victor, Jeff, 300
- Viewing
  - control domains, 98–100
  - Dynamic Domains, 69–73
  - Logical Domains, 96–97, 100–101
- VIO (virtual I/O) VEs, 33, 82–83
- virt-install command, 273
- Virt-io device, 140
- Virtual console concentrators (vcc), 88
- Virtual CPUs (vCPUs), 71
  - Logical Domains, 78–80, 84–85
  - VirtualBox, 134–136
- Virtual Desktop Integration (VDI), 332
- Virtual disks
  - guests, 261
  - Logical Domains, 86–88
  - VirtualBox, 137–140
- Virtual environments (VEs), 1–2
  - asset management, 325–326
  - disaster recovery, 313–315
  - life cycle, 303–306
  - migration. *See* Migration
  - restoration, 314, 314
  - snapshots, 7, 20
- Virtual I/O (VIO) VEs, 33, 82–83
- virtual machine monitors (VMMs), 113–114, 131
- Virtual machines (VMs), 2
  - adoption of, 331–332

- emergence of, 330-331
- full virtualization and para-virtualization, 35
- guest. *See* Guests
- history, 328-329
- hypervisor relative strengths, 36
- industry examples, 33-34
- Type 1 hypervisors, 32-33
- Type 2 hypervisors, 34-35
- Virtual management
  - environment (VME), 33
- Virtual Media Manager, 140, 153-154, 158
- Virtual memory
  - Containers, 203-205
  - OSV, 41
  - workload consolidation, 11
- Virtual motherboards, 137
- Virtual Network Computing (VNC), 116, 261, 263, 265
- Virtual network devices
  - Logical Domains, 85-86
  - VirtualBox, 140-142
- Virtual network interface cards (VNICs), 222-223
- Virtual Network Terminal
  - Server daemon, 88
- Virtual networks for Containers, 222-225
- Virtual pools, 317-318
- Virtual-processor identifiers (VPIDs), 343
- Virtual Remote Desktop Protocol (VRDP) server, 161
- Virtual switches (vSwitches), 222
- VirtualBox. *See* Oracle VM VirtualBox
- VirtualBox command, 148
- .VirtualBox directory, 154
- VirtualBox process, 133
- Virtualization Controller (VC), 317
- Virtualization management, 301-302
  - business agility, 311-312
  - business continuity, 312-313
  - disaster recovery, 313-315
  - dynamic resource consumption, 307-308
  - dynamic workloads, 308-309
  - operational flexibility, 309-311
  - Ops Center, 315-326
    - physical computer
      - limitations, 307
    - summary, 326
    - VE life cycle, 303-306
- Virtualization overview, 1-2
  - benefits, 2-5
  - concepts, 8-9
  - reasons, 7-8
  - summary, 41-42
  - system virtualization
    - models. *See* System virtualization models
  - use cases. *See* Use cases
  - workload consolidation, 5-7
- Virtualization service client (VSC), 124
- Virtualization service provider (VSP), 124
- Virtualization technologies
  - comparisons, 227
  - Containers, 233-235
  - decision tree, 236-237
  - diverse OS consolidation, 239
  - Dynamic Domains, 230-231
  - IP web service hosting, 238-239
  - large workload consolidation, 237-238
  - Logical Domains, 231-232
  - requirements, 235-236
  - strengths and limitations, 227-230
  - summary, 239
  - x86 hypervisors, 232-233
- Virtualization Technology (VT)
  - processors, 116
- VM/370 systems, 33, 331, 336-337, 340-342
- VMBus, 124
- VMDK disk format, 23, 138
- VME (virtual management environment), 33
- VMMs (virtual machine monitors), 113-114, 131
- VMotion feature, 121
- VMs. *See* Virtual machines (VMs)
- vmstat command, 84, 207
- VMware ESX, 33-35, 121-122
  - drivers, 122-123
  - features, 122
  - support, 123
- VMware Infrastructure (VI), 121
- VMware scheduler, 337
- VNC (Virtual Network Computing), 116, 261, 263, 265
- vncviewer tool, 263, 265, 273
- VNICs (virtual network interface cards), 222-223
- vntsd service, 88
- VPIDs (virtual-processor identifiers), 343
- VRDP (Virtual Remote Desktop Protocol) server, 161
- VSC (virtualization service client), 124
- VSP (virtualization service provider), 124
- vSphere features, 122
- vSwitches (virtual switches), 222
- VT (Virtualization Technology)
  - processors, 116
- VT-X CPUs, 125, 130
- W**
- Warm migration, 21-22, 105
- Web service hosting, 238-239
- Whole-root Containers, 180
- Windows Management Instrumentation (WMI), 324
- Wizards for Ops Center, 321
- Workload consolidation, 5-7, 9-12
  - availability, 16-18
  - Containers for, 285-290
  - nodes in, 16
  - performance effects, 10, 305
  - resource controls, 12-16
  - virtualization technologies
    - for, 237-239
- Workloads
  - asynchronous, 18
  - bursty, 18-19
  - dynamic, 308-309
  - future, 8
  - isolation, 227-228
  - mobility, 20-23
  - relationships, 308
  - templates, 312
- X**
- x86 environments
  - hypervisor advantages, 232-233
- Oracle VM Server. *See* Oracle VM Server for x86
- as virtualization selection
  - factors, 237

- x86 guests, 113–114
  - Citrix XenServer, 120–121
  - Hardware Compatibility List for, 114
  - Microsoft Windows Server 2008 Hyper-V, 123–125
  - Oracle VM Server for x86, 116–118
  - Red Hat Enterprise Virtualization, 125–127
  - summary, 127
  - Type 1 hypervisors, 114–115
  - VMware ESX, 121–123
  - Xen project hypervisors, 115–116
    - xVM hypervisor, 118–119
- XBUs (crossbar board units), 46–47
- Xen project hypervisors, 115–116
- xend-relocation-address field, 274–275
- xend-relocation-hosts-allow field, 274–275
- xend-relocation-server property, 274–275
- XenServer, 34–35, 120–121
  - drivers, 120–121
  - features, 120
  - hypervisor, 34
    - support, 121
- XSBs, 49
  - Dynamic Domains, 70, 74–75
  - Quad-XSB, 51–56
  - Uni-XSB, 50–51
- XSCF (Extended System Control Facility), 44, 73–74
- xVM hypervisor, 34
  - configuration, 269–272
  - live migration. *See* Live migration
  - overview, 118–119
- Z**
- z/OS operating system, 33
- z/VM hypervisor, 33
- ZFS files systems, 86–87
  - Container cloning, 217
  - Container storage, 185
  - domain cloning, 103
  - guest cloning, 164–165
  - Microsoft Windows in Containers, 277–278
  - VirtualBox, 146
- zfs snapshot command, 103
- zlogin command, 187–188, 199–200, 224, 289, 295
- zoneadm command for Containers
  - booting, 296
  - cloning, 216–217
  - halting, 188–189
  - ID numbers, 198
  - installing, 186–188, 279, 288
  - migrating, 217–218
  - virtual memory, 204
- zonecfg command for Containers
  - cloning, 216
  - configuring, 181
  - NICs, 213–214
  - resource management, 189–191, 201–202
  - virtual memory, 205–206
- zonecfg set limitpriv command, 294
- zonecfgm command, 279
- zonename property, 182
- zonepath command, 217–219
- zonepath property, 182
- Zones
  - containers. *See* Oracle Solaris Containers
  - VirtualBox, 145
    - sysidcfg file, 188
- zonestat tool, 211
- zsched process, 199–200
- zvmstat script, 207, 212