



Logical Domains (LDoms) 1.0.3

管理指南

Sun Microsystems, Inc.
www.sun.com

文件号码 820-5005-10
2008 年 6 月, 修订版 A

请将有关本文档的意见和建议提交至: <http://www.sun.com/hwdocs/feedback>

版权所有 © 2008 Sun Microsystems, Inc., 4150 Network Circle, Santa Clara, California 95054, U.S.A. 保留所有权利。

对于本文档中介绍的产品，Sun Microsystems, Inc. 对其所涉及的技术拥有相关的知识产权。需特别指出的是（但不局限于此），这些知识产权可能包含在 <http://www.sun.com/patents> 中列出的一项或多项美国专利，以及在美国和其他国家/地区申请的一项或多项其他专利或待批专利。

美国政府权利 - 商业软件。政府用户应遵循 Sun Microsystems, Inc. 的标准许可协议，以及 FAR（Federal Acquisition Regulations，即“联邦政府采购法规”）的适用条款及其补充条款。

本产品的某些部分可能是从 Berkeley BSD 系统衍生出来的，并获得了加利福尼亚大学的许可。UNIX 是 X/Open Company, Ltd. 在美国和其他国家/地区独家许可的注册商标。

Sun、Sun Microsystems、Sun 徽标、Java、Solaris、JumpStart、OpenBoot、Sun Fire、Netra、SunSolve、Sun BluePrints、Sun Blade、Sun Ultra 和 SunVTS 是 Sun Microsystems, Inc. 在美国和其他国家/地区的服务标记、商标或注册商标。

所有 SPARC 商标的使用均已获得许可，它们是 SPARC International, Inc. 在美国和其他国家/地区的商标或注册商标。标有 SPARC 商标的产品均基于由 Sun Microsystems, Inc. 开发的体系结构。

Adobe PostScript 徽标是 Adobe Systems, Incorporated 的商标。

本服务手册所介绍的产品以及所包含的信息受美国出口控制法制约，并应遵守其他国家/地区的进出口法律。严禁将本产品直接或间接地用于核设施、导弹、生化武器或海上核设施，也不能直接或间接地出口给核设施、导弹、生化武器或海上核设施的最终用户。严禁出口或转口到美国禁运的国家/地区以及美国禁止出口清单中所包含的实体，包括但不限于被禁止的个人以及特别指定的国家/地区的公民。

本文档按“原样”提供，对于所有明示或默示的条件、陈述和担保，包括对适销性、适用性或非侵权性的默示保证，均不承担任何责任，除非此免责声明的适用范围在法律上无效。



Adobe PostScript

目录

前言 xvii

1. Logical Domains 软件概述 1

虚拟机管理程序和逻辑域 1

Logical Domains Manager 3

Logical Domains 的角色 3

命令行界面 4

虚拟输入/输出 4

虚拟网络 4

虚拟存储 5

虚拟控制台 5

动态重新配置 5

延迟重新配置 5

持久性配置 6

2. 安全性 7

安全注意事项 7

Solaris Security Toolkit 和 Logical Domains Manager 8

强化 8

最小化逻辑域 10

- 授权 10
- 审计 11
- 遵从性 11

3. 安装和启用软件 13

- 升级 Solaris OS 13

- 保存和恢复 Logical Domains 约束数据库文件 13

- 对控制域使用即时升级 14

- 升级到 LDoms 1.0.3 软件 14

- ▼ 从 LDoms 1.0 升级到 LDoms 1.0.3 软件 14

- 在控制域上首次安装软件 15

- ▼ 安装 Solaris 10 OS 16

- ▼ 升级系统固件 17

- ▼ 在不使用 FTP 服务器的情况下升级系统固件 18

- ▼ 降级系统固件 18

- 下载 Logical Domains Manager 和 Solaris Security Toolkit 19

- ▼ 下载 Logical Domains Manager、Solaris Security Toolkit 和 Logical Domains MIB 19

- 安装 Logical Domains Manager 和 Solaris Security Toolkit 20

- 使用安装脚本安装 Logical Domains Manager 1.0.3 和 Solaris Security Toolkit 4.2 软件 20

- ▼ 使用不带任何选项的 `install-ldm` 脚本进行安装 21

- ▼ 使用带有 `-d` 选项的 `install-ldm` 脚本进行安装 24

- ▼ 使用带有 `-d none` 选项的 `install-ldm` 脚本进行安装 25

- ▼ 使用带有 `-p` 选项的 `install-ldm` 脚本进行安装 26

- 使用 JumpStart 安装 Logical Domains Manager 1.0.3 和 Solaris Security Toolkit 4.2 软件 26

- ▼ 安装 JumpStart 服务器 27

- ▼ 使用 JumpStart 软件进行安装 27

手动安装 Logical Domains Manager 和 Solaris Security Toolkit 软件	29
▼ 手动安装 Logical Domains Manager (LDoms) 1.0.3 软件	29
▼ (可选) 手动安装 Solaris Security Toolkit 4.2 软件	30
▼ (可选) 手动强化控制域	30
▼ 验证强化	31
▼ 撤消强化	31
启用 Logical Domains Manager 守护进程	31
▼ 启用 Logical Domains Manager 守护进程	32
创建授权和配置文件并为用户帐户分配角色	32
管理用户授权	33
▼ 为用户添加授权	33
▼ 为用户删除所有授权	33
管理用户配置文件	33
▼ 为用户添加配置文件	34
▼ 为用户删除所有配置文件	34
为用户分配角色	34
▼ 创建角色并将该角色分配给用户	34
4. 设置服务和逻辑域	37
输出消息	37
Sun UltraSPARC T1 处理器	37
Sun UltraSPARC T2 处理器	38
创建默认服务	38
▼ 创建默认服务	38
控制域的初始配置	40
▼ 设置控制域	40
重新引导以使用逻辑域	42
▼ 重新引导	42

启用控制/服务域与其他域之间的联网 42

▼ 将虚拟交换机配置为主接口 43

启用虚拟网络终端服务器守护进程 44

▼ 启用虚拟网络终端服务器守护进程 44

创建和启动来宾域 45

▼ 创建和启动来宾域 45

对来宾域进行 JumpStart 设置 48

5. 将虚拟磁盘与 **Logical Domains** 结合使用 49

虚拟磁盘简介 49

管理虚拟磁盘 50

▼ 添加虚拟磁盘 50

▼ 多次导出虚拟磁盘后端 51

▼ 更改虚拟磁盘选项 51

▼ 更改超时选项 52

▼ 删除虚拟磁盘 52

虚拟磁盘外观 52

完整磁盘 52

具有单个片的磁盘 53

虚拟磁盘后端选项 53

只读 (ro) 选项 53

独占 (excl) 选项 53

片 (slice) 选项 54

虚拟磁盘后端 54

物理磁盘或磁盘 LUN 55

▼ 将物理磁盘导出为虚拟磁盘 55

物理磁盘片 56

▼ 将物理磁盘片导出为虚拟磁盘 56

▼ 导出片 2 56

文件和卷	57
导出为完整磁盘的文件或卷	57
▼ 将文件导出为完整磁盘	57
导出为具有单个片的磁盘的文件或卷	58
▼ 将 ZFS 卷导出为具有单个片的磁盘	58
导出卷以及向后兼容性	59
不同类型后端的导出方式汇总	59
原则	60
CD、DVD 和 ISO 映像	60
▼ 将 CD 或 DVD 从服务域导出到来宾域	61
虚拟磁盘超时	62
虚拟磁盘和 SCSI	63
虚拟磁盘和 format(1M) 命令	63
将 ZFS 用于虚拟磁盘	64
在 ZFS 卷之上创建虚拟磁盘	64
▼ 在 ZFS 卷之上创建虚拟磁盘	64
在虚拟磁盘上使用 ZFS	65
▼ 在虚拟磁盘上使用 ZFS	66
将 ZFS 用于引导磁盘	67
▼ 将 ZFS 用于引导磁盘	68
在逻辑域环境中使用卷管理器	69
在卷管理器之上使用虚拟磁盘	69
在 SVM 之上使用虚拟磁盘	71
在安装了 VxVM 的情况下使用虚拟磁盘	71
在虚拟磁盘之上使用卷管理器	72
在虚拟磁盘之上使用 ZFS	72
在虚拟磁盘之上使用 SVM	73
在虚拟磁盘之上使用 VxVM	73

6. 其他信息和任务 75

在 CLI 中输入名称的限制 75

文件名称 (*file*) 和变量名称 (*var_name*) 75

虚拟磁盘服务器 *backend* 和虚拟交换机设备名称 75

配置名称 (*config_name*) 75

所有其他名称 76

使用 `ldm list` 子命令 76

计算机可读的输出 76

▼ 显示 `ldm` 子命令的语法用法 76

标志定义 79

使用率统计信息定义 80

各种列表示例 80

▼ 显示软件版本 (`-v`) 80

▼ 生成短列表 80

▼ 生成长列表 (`-l`) 81

▼ 生成扩展列表 (`-e`) 82

▼ 生成可解析的、计算机可读的列表 (`-p`) 84

▼ 显示域的状态 84

▼ 列出变量 85

▼ 列出绑定 85

▼ 列出配置 86

▼ 列出设备 86

▼ 列出服务 88

列出约束 88

▼ 列出一个域的约束 88

▼ 以 XML 格式列出约束 89

▼ 以计算机可读格式列出约束 90

ldm stop-domain 命令在域负载较重时会超时	91
确定与虚拟网络设备对应的 Solaris 网络接口名称	91
▼ 查找 Solaris OS 网络接口名称	92
自动或手动分配 MAC 地址	93
分配给 Logical Domains 软件的 MAC 地址范围	93
自动分配算法	94
检测重复的 MAC 地址	94
释放的 MAC 地址	95
CPU 和内存地址映射	95
CPU 映射	95
▼ 确定 CPU 编号	96
内存映射	96
▼ 确定实际内存地址	96
CPU 和内存映射示例	96
配置分离式 PCI Express 总线以使用多个逻辑域	98
▼ 创建分离式 PCI 配置	99
在 PCI 总线上启用 I/O MMU 旁路模式	101
使用控制台组	102
▼ 将多个控制台组成一个组	102
将逻辑域从一个服务器移动至另一个服务器	103
▼ 设置域以便移动	103
▼ 移动域	103
删除逻辑域	103
▼ 删除所有来宾逻辑域	104
在逻辑域中操作 Solaris OS	105
如果启用了域，则在启动 Solaris OS 之后，OpenBoot 固件不可用	105
对服务器执行关开机循环	105
▼ 将当前逻辑域配置保存到 SC	105

OpenBoot power-off 命令的结果	106
在 Solaris OS 中发出中断的结果	106
停止或重新引导控制域的结果	106
将 LDoms 与 ALOM CMT 配合使用	108
▼ 将逻辑域配置重置为默认配置或其他配置	108
启用并使用 BSM 审计	109
▼ 使用 enable-bsm.fin 结束脚本	109
▼ 使用 Solaris OS bsmconv(1M) 命令	110
▼ 检验 BSM 审计是否已启用	110
▼ 禁用审计	110
▼ 列显审计输出	111
▼ 轮转审计日志	111
支持的网络适配器	111
▼ 确定网络适配器是否与 GLDv3 兼容	111
针对 NAT 和路由配置虚拟交换机和服务域	112
▼ 设置虚拟交换机以实现与域的外部连接	112
在 Logical Domains 环境中配置 IPMP	113
在逻辑域中将虚拟网络设备配置到 IPMP 组中	113
在服务域中配置并使用 IPMP	114
词汇表	115



-
- 图 1-1 支持两个逻辑域的虚拟机管理程序 2
- 图 5-1 虚拟磁盘与 Logical Domains 50
- 图 6-1 两个连接到不同虚拟交换机实例的虚拟网络 113
- 图 6-2 连接到不同服务域的各个虚拟网络设备 114
- 图 6-3 两个配置为属于 IPMP 组的网络接口 114

表

表 1-1	逻辑域的角色	3
表 2-1	ldm 子命令及用户授权	10
表 6-1	停止或重新引导控制 (primary) 域的预期行为	107

代码示例

代码示例 3-1	下载的 Logical Domains 1.0.3 软件的目录结构	19
代码示例 3-2	LDoms 强化 Solaris 配置的输出	22
代码示例 3-3	选择自定义配置文件情况下的输出	23
代码示例 3-4	install-ldm -d 脚本运行成功时的输出	24
代码示例 3-5	install-ldm -d none 脚本运行成功时的输出	25
代码示例 6-1	所有 ldm 子命令的语法用法	76
代码示例 6-2	安装的软件版本	80
代码示例 6-3	所有域的短列表	80
代码示例 6-4	所有域的长列表	81
代码示例 6-5	所有域的扩展列表	82
代码示例 6-6	计算机可读的列表	84
代码示例 6-7	域状态	84
代码示例 6-8	域的变量列表	85
代码示例 6-9	域的绑定列表	85
代码示例 6-10	配置列表	86
代码示例 6-11	所有服务器资源列表	86
代码示例 6-12	服务列表	88
代码示例 6-13	一个域的约束列表	88
代码示例 6-14	某个域的约束（采用 XML 格式）	89
代码示例 6-15	所有域的约束（采用计算机可读的格式）	90
代码示例 6-16	逻辑域配置的可解析长列表	97

前言

《Logical Domains (LDoms) 1.0.3 管理指南》提供了有关在支持的服务器、刀片和服务模块上运行的 Logical Domains Manager 1.0.3 软件的详细信息，包括概述和安全注意事项方面的信息，以及安装、配置、修改和执行常规任务的具体过程。请参阅《Logical Domains (LDoms) 1.0.3 发行说明》中的“支持的平台”，了解相关列表。本指南面向这些服务器的系统管理员，他们具有 UNIX® 系统和 Solaris™ 操作系统 (Solaris™ Operating System, Solaris OS) 的相关知识。

阅读本书之前

如果您不具有 UNIX 命令和操作过程以及 Solaris 操作系统的相关知识，请阅读随系统硬件一同提供的 Solaris OS 用户和系统管理员文档，并考虑参加 UNIX 系统管理培训。

本书的结构

第 1 章对 Logical Domains 软件进行了概述。

第 2 章介绍了 Solaris Security Toolkit，以及它如何为逻辑域中的 Solaris OS 提供安全。

第 3 章介绍了升级或安装以及启用 Logical Domains Manager 软件的详细过程。

第 4 章介绍了设置服务和逻辑域的详细过程。

第 5 章介绍了如何将虚拟磁盘与 Logical Domains 软件结合使用。

第 6 章介绍了使用 Logical Domains 软件管理逻辑域时执行常规任务的其他信息和过程。

词汇表列出了特定于 LDoms 的缩写、首字母缩略词和术语及其定义。

使用 UNIX 命令

本文档不会介绍基本的 UNIX 命令和操作过程，如关闭系统、启动系统和配置设备等。欲获知此类信息，请参阅以下文档：

- 系统附带的软件文档
- Solaris 操作系统的有关文档，其 URL 如下：
`http://docs.sun.com`

Shell 提示符

Shell	提示符
C shell	<i>machine-name%</i>
C shell 超级用户	<i>machine-name#</i>
Bourne shell 和 Korn shell	\$
Bourne shell 和 Korn shell 超级用户	#

印刷约定

字体*	含义	示例
AaBbCc123	命令、文件和目录的名称；计算机屏幕输出	编辑 .login 文件。 使用 ls -a 列出所有文件。 % You have mail.
AaBbCc123	用户键入的内容，与计算机屏幕输出的显示不同	% su Password:
AaBbCc123	保留未译的新词或术语以及要强调的词。要使用实名或值替换的命令行变量。	这些称为 class 选项。 要删除文件，请键入 rm filename。
新词术语强调	新词或术语以及要强调的词。	您 必须 成为超级用户才能执行此操作。
《书名》	书名	阅读《用户指南》的第 6 章。

* 浏览器的设置可能会与这些设置有所不同。

相关文档

您可以从以下位置获得《Logical Domains (LDoms) 1.0.3 管理指南》和《Logical Domains (LDoms) 1.0.3 发行说明》：

<http://docs.sun.com>

您可以在以下 Sun BluePrints™ 站点找到《Beginners Guide to LDoms: Understanding and Deploying Logical Domains Software》：

<http://www.sun.com/blueprints/0207/820-0832.html>

您可以在以下位置找到与服务器、软件或 Solaris OS 相关的文档：

<http://docs.sun.com>

在搜索框中键入服务器、软件或 Solaris OS 的名称，即可找到所需的文档。

应用	书名	文件号码	格式	所在位置
LDoms 发行说明	《Logical Domains (LDoms) 1.0.3 发行说明》	820-5011-10	HTML PDF	联机提供
LDoms 的 Solaris 手册页	Solaris 10 Reference Manual Collection: <ul style="list-style-type: none">• drd(1M) 手册页• vntsd(1M) 手册页	N/A	HTML	联机提供
LDoms 手册页	ldm(1M) 手册页	N/A	SGML	联机提供
	《Logical Domains (LDoms) 1.0.1 Manager Man Page Guide》	820-4896-10	PDF	联机提供
Logical Domains 软件基本知识	《Beginners Guide to LDoms: Understanding and Deploying Logical Domains Software》	820-0832-20	PDF	联机提供
LDoms MIB 管理	《Logical Domains (LDoms) MIB 1.0.1 管理指南》	820-3458-10	HTML PDF	联机提供
LDoms MIB 发行说明	《Logical Domains (LDoms) MIB 1.0.1 发行说明》	820-3464-10	HTML PDF	联机提供
包括安装、使用 JumpStart™ 和使用 SMF 在内的 Solaris OS	Solaris 10 Collection	N/A	HTML PDF	联机提供
安全性	《Solaris Security Toolkit 4.2 管理指南》	819-3791-10	HTML PDF	联机提供
安全性	《Solaris Security Toolkit 4.2 Reference Manual》	819-1503-10	HTML PDF	联机提供
安全性	《Solaris Security Toolkit 4.2 发行说明》	819-3798-10	HTML PDF	联机提供
安全性	《Solaris Security Toolkit 4.2 Man Page Guide》	819-1505-10	HTML PDF	联机提供

文档、支持和培训

Sun 提供的服务	URL
文档	http://docs.sun.com
支持	http://www.sun.com/support
培训	http://www.sun.com/training

第三方 Web 站点

Sun 对本文档中提到的第三方 Web 站点的可用性不承担任何责任。对于此类站点或资源中的（或通过它们获得的）任何内容、广告、产品或其他资料，Sun 并不表示认可，也不承担任何责任。对于因使用或依靠此类站点或资源中的（或通过它们获得的）任何内容、产品或服务而造成的或连带产生的实际或名义损坏或损失，Sun 概不负责，也不承担任何责任。

Sun 欢迎您提出意见

Sun 致力于提高其文档的质量，并十分乐意收到您的意见和建议。您可以通过以下网址提交您的意见和建议：

<http://www.sun.com/hwdocs/feedback>

请在您的反馈信息中包含文档的书名和文件号码：

《Logical Domains (LDoms) 1.0.3 管理指南》，文件号码 820-5005-10。

第1章

Logical Domains 软件概述

本章对 Logical Domains 软件进行了简要概述。Solaris 10 11/06 发行版（最低要求）以及必需的修补程序提供了使用 Sun 的 Logical Domains 技术所必需的全部 Solaris OS 功能。但是，要使用逻辑域，还需要系统固件和 Logical Domains Manager。有关具体详细信息，请参阅《Logical Domains (LDoms) 1.0.3 发行说明》中的“必需的和推荐的软件”。

虚拟机管理程序和逻辑域

本节简要介绍了 SPARC® 虚拟机管理程序以及它所支持的逻辑域。

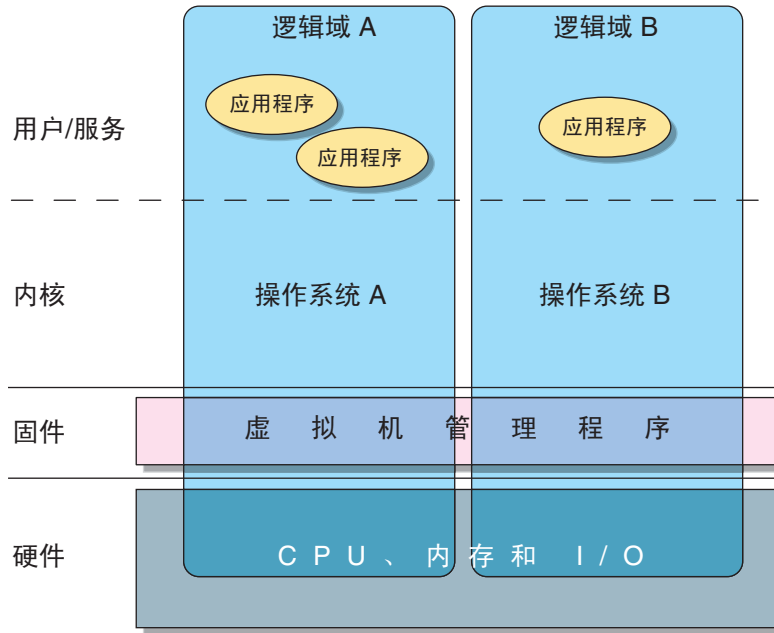
SPARC 虚拟机管理程序是一个小固件层，提供了可以向其中写入操作系统的稳定虚拟机体系结构。使用虚拟机管理程序的 Sun 服务器提供的硬件功能，支持虚拟机管理程序对逻辑操作系统活动的控制。

逻辑域是一个在单个计算机系统内具有自己的操作系统、资源和身份的离散逻辑分组。可以单独创建、销毁、重新配置及重新引导每个逻辑域，而无需关闭再打开服务器电源。可以在不同的逻辑域中运行各种应用程序软件，并使其保持相互独立，以获得相应的性能和安全。

每个逻辑域只能发现由虚拟机管理程序设置为可用的那些服务器资源并与之交互。使用 Logical Domains Manager，系统管理员可以指定虚拟机管理程序要通过控制域执行的操作。这样，虚拟机管理程序会执行服务器资源的划分，并向多个操作系统环境提供有限的子集。这就是创建逻辑域的基本机制。下图显示了支持两个逻辑域的虚拟机管理程序。它还显示了构成 Logical Domains 功能的各层：

- 应用程序，即用户/服务
- 内核，即操作系统
- 固件，即虚拟机管理程序
- 硬件，包括 CPU、内存和 I/O

图 1-1 支持两个逻辑域的虚拟机管理程序



特定的 SPARC 虚拟机管理程序所支持的逻辑域的数量和功能是与服务器相关的特性。虚拟机管理程序可以向给定的逻辑域分配服务器的全部 CPU、内存和 I/O 资源的子集。这样便可以同时支持多个操作系统，每个操作系统位于自己的逻辑域内。可以使用任意粒度在各个逻辑域之间重新安排资源。例如，可以使用 8 千字节的粒度将内存分配到逻辑域中。

每个虚拟机可以作为具有自己的如下资源的一个完全独立的机器进行管理：

- 内核、修补程序和调节参数
- 用户帐户和管理员
- 磁盘
- 网络接口、MAC 地址和 IP 地址

可以独立地停止、启动和重新引导每个虚拟机，而无需对服务器执行关开机循环。

虚拟机管理程序软件负责保持各个逻辑域之间的相互独立。虚拟机管理程序软件还提供了逻辑域通道 (logical domain channel, LDC)，因此逻辑域可以相互通信。使用逻辑域通道，域可以相互提供服务，例如联网或磁盘服务。

系统控制器可以监视和运行物理机，但不管理虚拟机。Logical Domains Manager 可以运行虚拟机。

Logical Domains Manager

Logical Domains Manager 可用于创建和管理逻辑域。每个服务器只能有一个 Logical Domains Manager。Logical Domains Manager 可将逻辑域映射为物理资源。

Logical Domains 的角色

所有的逻辑域都是相同的，只是为它们指定的角色不同。逻辑域可以承担多种角色。

表 1-1 逻辑域的角色

域角色	说明
控制域	Logical Domains Manager 在此域中运行，允许您创建和管理其他逻辑域，并向其他域分配虚拟资源。每台服务器只能有一个控制域。安装 Logical Domains 软件时创建的第一个域是控制域，它名为 <code>primary</code> 。
服务域	为其他域提供虚拟设备（如虚拟交换机、虚拟控制台集线器和虚拟磁盘服务器）服务的域。
I/O 域	此域对物理 I/O 设备（例如 PCI Express 控制器中的网卡）具有直接拥有权和直接访问权。如果 I/O 域也是控制域，则 I/O 域与其他域共享以虚拟设备形式出现的设备。您可以拥有的 I/O 域的数量取决于您的平台体系结构。例如，如果您使用的是 Sun UltraSPARC® T1 处理器，最多可以拥有两个 I/O 域，其中一个 I/O 域还必须是控制域。
来宾域	此域由控制域管理，并使用来自 I/O 域和服务域的服务。

如果您具备一个现有系统，并且服务器上正在运行操作系统和其他软件，则在安装 Logical Domains Manager 后，这便成为您的控制域。设立控制域后，您可能需要从控制域中移除一些应用程序，并平衡各个域中的应用程序负载，从而最有效地利用系统。

命令行界面

Logical Domains Manager 提供了一个命令行界面 (command-line interface, CLI)，供系统管理员创建和配置逻辑域时使用。CLI 是具有多个子命令的单个命令 `ldm(1M)`。

要使用 Logical Domains Manager CLI，必须运行 Logical Domains Manager 守护进程 `ldmd`。有关 `ldm(1M)` 命令及其子命令的详细说明，请参见 `ldm(1M)` 手册页和 Logical Domains (LDoms) Manager 手册页指南。`ldm(1M)` 手册页是 `SUNWldm` 软件包的一部分，并在安装 `SUNWldm` 软件包时进行安装。

要执行 `ldm` 命令，必须在 UNIX `$PATH` 变量中指定 `/opt/SUNWldm/bin` 目录。要访问 `ldm(1M)` 手册页，请将目录路径 `/opt/SUNWldm/man` 添加到 `$MANPATH` 变量中。执行上述两个操作所用的命令如下所示：

```
$ PATH=$PATH:/opt/SUNWldm/bin; export PATH (对于 Bourne 或 K shell)
$ MANPATH=$MANPATH:/opt/SUNWldm/man; export MANPATH
% set PATH=($PATH /opt/SUNWldm/bin) (对于 C shell)
% set MANPATH=($MANPATH /opt/SUNWldm/man)
```

虚拟输入/输出

在 Logical Domains 环境中，管理员可在 Sun Fire™ 或 SPARC Enterprise T1000 或 T2000 服务器上置备多达 32 个域。虽然可以为每个域分配专用的 CPU 和内存，但是由于这些系统中的 I/O 总线和物理 I/O 插槽的数量有限，因此无法向所有域提供对磁盘和网络设备的独占访问。虽然可以通过将 PCI Express® (PCI-E) 总线一分为二来共享一些物理设备（请参见第 98 页中的“配置分离式 PCI Express 总线以使用多个逻辑域”），但也不足以向所有域提供独占的设备访问。通过实现虚拟化 I/O 模型，可以解决物理 I/O 设备直接访问不足的问题。

所有不能进行直接 I/O 访问的逻辑域都可配置虚拟 I/O 设备，这些虚拟设备与服务域进行通信，可通过服务域所运行的服务来提供对物理设备或其功能的访问。在此客户机-服务器模型中，虚拟 I/O 设备通过称为逻辑域通道 (logical domain channel, LDC) 的域际通信通道相互通信或与对应服务通信。在 Logical Domains 1.0.3 软件中，虚拟化 I/O 功能包含对虚拟网络、存储和控制台的支持。

虚拟网络

使用虚拟网络设备和虚拟网络交换机设备这两个组件即可实现虚拟网络支持。虚拟网络 (vnet) 设备可模仿以太网设备，并使用点对点通道与系统中的其他 vnet 设备进行通信。虚拟交换机 (vsw) 设备主要充当所有虚拟网络的传入和传出包的多路复用器。vsw 设备可直接与服务域上的物理网络适配器通信，并代表虚拟网络发送和接收包。vsw 设备还充当简单第二层交换机，并在系统内与 vsw 设备连接的 vnet 设备之间交换包。

虚拟存储

借助虚拟存储基础结构，逻辑域可以通过客户机-服务器模型访问未直接分配给它们的块级存储。虚拟存储基础结构由两个组件组成：虚拟磁盘客户机 (vdc) 和虚拟磁盘服务 (vds)。前者作为块设备接口导出，后者代表虚拟磁盘客户机处理磁盘请求，并将这些请求提交给服务域上的物理存储。虽然虚拟磁盘显示为客户机域上的常规磁盘，但是所有磁盘操作都会通过虚拟磁盘服务转发到物理磁盘。

虚拟控制台

在 Logical Domains 环境中，除 primary 域之外的所有域的控制台 I/O 会重定向到运行虚拟控制台集中器 (vcc) 和虚拟网络终端服务器的服务域，而非系统控制器。虚拟控制台集中器服务可以充当所有域的控制台通信流量的集中器，还可以与虚拟网络终端服务器守护进程 (vntsd) 通信，并通过 UNIX 套接字提供对每个控制台的访问。

动态重新配置

动态重新配置 (dynamic reconfiguration, DR) 是在操作系统运行的同时添加或删除资源的能力。Solaris 10 OS 仅支持添加和删除虚拟 CPU (vcpu)。Solaris 10 OS 不支持对内存和输入/输出进行动态重新配置。要在 Logical Domains Manager CLI 中使用动态重新配置功能，必须在要更改的域中运行 Logical Domains 动态重新配置守护进程 drd(1M)。

延迟重新配置

与可以立即发生的动态重新配置操作不同，延迟重新配置操作在下次重新引导 OS 之后生效，或者如果没有运行 OS，则在停止并启动逻辑域之后生效。对活动逻辑域执行的任何添加或删除操作（add-vcpu、set-vcpu 和 remove-vcpu 子命令除外）将被视为延迟重新配置操作。另外，对活动逻辑域执行 set-vswitch 子命令将被视为延迟重新配置操作。

如果您使用的是 Sun UltraSPARC T1 处理器，则首次安装并启用 Logical Domains Manager 时或者将配置恢复为 factory-default 时，LDoms Manager 将以配置模式运行。在此模式下，会接受重新配置请求并将其排入队列，但不遵照执行。这将允许生成一个新配置，并将其存储到 SC，而不影响运行中的机器的状态，因此，也不会因延迟重新配置和重新引导 I/O 域等操作的任何限制而受阻碍。

一旦特定逻辑域的延迟重新配置正在进行中，对该逻辑域的任何其他重新配置请求也将延迟，直到重新引导域或者停止并启动域。此外，如果一个逻辑域有未完成的延迟重新配置，则对其他逻辑域的重新配置请求会受到严格限制，而且会失败并显示相应的错误消息。

即使将试图移除活动逻辑域上的虚拟 I/O 设备作为延迟重新配置操作进行处理，某些配置更改也会立即发生。这意味着，相关的 Logical Domains Manager CLI 操作一经调用，该设备实际上将立即停止运行。

Logical Domains Manager 子命令 `remove-reconf` 可取消延迟重新配置操作。可以使用 `ldm list-domain` 命令列出延迟重新配置操作。有关如何使用延迟重新配置功能的更多信息，请参阅 `ldm(1M)` 手册页或 Logical Domains (LDom)s Manager 手册页指南。

注 – 如果对虚拟 I/O 设备发出了任何其他 `ldm remove-*` 命令，则不能使用 `ldm remove-reconf` 命令。在这种情况下，`ldm remove-reconf` 命令将失败。

持久性配置

使用 Logical Domains Manager CLI 命令可以将逻辑域的当前配置存储在系统控制器 (system controller, SC) 上。可以添加配置、指定要使用的配置、删除配置，并列出系统控制器上的配置。（请参阅 `ldm(1M)` 手册页或 Logical Domains (LDom)s Manager 手册页指南。）另外，还有一个 ALOM CMT 1.3 版命令，通过该命令可以选择要引导的配置（请参见第 108 页中的“将 LDom)s 与 ALOM CMT 配合使用”）。

第2章

安全性

本章介绍 Solaris Security Toolkit 软件，以及如何使用它保护逻辑域中的 Solaris OS 的安全。

安全注意事项

Solaris Security Toolkit 软件（非正式地称为 JumpStart™ 体系结构和安全脚本 (JumpStart™ Architecture and Security Script, JASS) 工具包）提供了一种用于构建和维护安全的 Solaris OS 系统的自动、可扩展和可伸缩的机制。Solaris Security Toolkit 可以为在服务器管理（包括 Logical Domains Manager 中的控制域）中起着关键作用的设备提供安全性。

Solaris Security Toolkit 4.2 软件包 (SUNWjass) 通过按以下方式使用 `install-ldm` 脚本，提供了保护控制域上的 Solaris 操作系统的多种方法：

- 使用 Logical Domains Manager 安装脚本 (`install-ldm`) 和特定于 Logical Domains Manager 的控制驱动程序 (`ldm_control-secure.driver`)，让 Solaris Security Toolkit 自动强化控制域。
- 在使用安装脚本时选择备用驱动程序。
- 在使用安装脚本和应用您自己的 Solaris 强化时，不选择任何驱动程序。

SUNWjass 软件包与 Logical Domains (LDoms) Manager 1.0.3 软件包 (SUNWldm) 一起位于 Sun 软件下载 Web 站点上。您可以选择在下载并安装 Logical Domains Manager 1.0.3 软件的同时，下载并安装 Solaris Security Toolkit 4.2 软件包。Solaris Security Toolkit 4.2 软件包中提供了使 Solaris Security Toolkit 软件能够与 Logical Domains Manager 配合工作所必需的修补程序。安装 Solaris Security Toolkit 4.2 软件后，便可以使用该软件强化您的系统。第 3 章介绍了如何安装和配置 Solaris Security Toolkit 并强化您的控制域。

以下安全功能是由 Solaris Security Toolkit 提供的，可供 Logical Domains Manager 用户使用：

- **强化** – 使用 Solaris Security Toolkit 4.2 软件（带有使 Solaris Security Toolkit 能够与 Logical Domains Manager 配合工作所必需的修补程序）修改 Solaris OS 配置以提高系统的安全性。
- **最小化** – 安装支持 LDoms 和 LDoms Management Information Base (MIB) 所必需的最少数量的核心 Solaris OS 软件包。
- **授权** – 使用适用于 Logical Domains Manager 的 Solaris OS 基于角色的访问控制 (Role-Based Access Control, RBAC) 设置授权。
- **审计** – 使用适用于 Logical Domains Manager 的 Solaris OS 基本安全模块 (Basic Security Module, BSM) 识别系统安全更改源，以确定执行了哪些操作、完成操作的时间、操作执行者以及产生了哪些影响。
- **遵从性** – 使用 Solaris Security Toolkit 的审计功能确定系统配置是否符合预定义的安全配置文件。

Solaris Security Toolkit 和 Logical Domains Manager

第 3 章介绍了如何安装 Solaris Security Toolkit，以使其能够与 Logical Domains Manager 配合工作。您可以在控制域上安装 Solaris Security Toolkit，控制域也是运行 Logical Domains Manager 的位置。您也可以在其他逻辑域上安装 Solaris Security Toolkit。二者唯一的区别是，前者需要使用 `ldm_control-secure.driver` 强化控制域，后者需要使用另一个驱动程序（例如 `secure.driver`）强化其他逻辑域。这是因为 `ldm_control-secure.driver` 是专用于控制域的。`ldm_control-secure.driver` 基于 `secure.driver`，而且已经为与 Logical Domains Manager 配合工作进行了自定义和测试。有关 `secure.driver` 的更多信息，请参阅《Solaris Security Toolkit 4.2 Reference Manual》。

强化

Solaris Security Toolkit 用于在控制域上强化 Solaris OS 的驱动程序 (`ldm_control-secure.driver`) 是专门定制的，以便 Logical Domains Manager 能够与 OS 协同运行。`ldm_control-secure.driver` 与《Solaris Security Toolkit 4.2 Reference Manual》中介绍的 `secure.driver` 类似。

`ldm_control-secure.driver` 为运行 Logical Domains Manager 软件的系统的控制域提供了基准配置。它旨在为 Solaris OS 域提供比典型数目更少的系统服务，保留控制域专供 Logical Domains Manager 操作使用，而不是用于一般用途。

如果尚未安装 Logical Domains Manager 软件，install-ldm 脚本会安装该软件并启用它。

下面简短概括了 secure.driver 的其他值得注意的变化。

- 禁止运行 Telnet 服务器。您可以改为使用安全 Shell (ssh)。此外，您仍可以使用 Telnet 客户端访问由 Logical Domains 虚拟网络终端服务器守护进程 (vntsd) 启动的虚拟控制台。例如，如果所运行的虚拟控制台正在侦听本地系统上的 TCP 端口 5001，则可以按如下方式访问该虚拟控制台。

```
# telnet localhost 5001
```

有关启用 vntsd 的说明，请参见第 31 页中的“启用 Logical Domains Manager 守护进程”。它不是自动启用的。

- 添加了下列结束脚本。这些脚本使得 Logical Domains Manager 可以安装和启动。在这些添加的脚本中，有些脚本必须添加到您创建的任何自定义驱动程序中，有些脚本则可选。这些脚本根据其是必需的还是可选的进行了标记。
 - install-ldm.fin - 安装 SUNWldm 软件包。（必需）
 - enable-ldmd.fin - 启用 Logical Domains Manager 守护进程 (ldmd)（必需）
 - enable-ssh-root-login.fin - 使超级用户可以通过安全 Shell (ssh) 直接登录。（可选）
- 更改了下列文件。这些更改对于您的任何自定义驱动程序都是可选的，因此标记为可选。
 - /etc/ssh/sshd_config - 允许超级用户帐户访问整个网络。在任一驱动程序中都不使用此文件。（可选）
 - /etc/ipf/ipf.conf - 公开了 UDP 端口 161 (SNMP)。（可选）
 - /etc/host.allow - 安全 Shell 守护进程 (sshd) 已面向整个网络（而不只是本地子网）公开。（可选）
- 已禁用（注释掉）下列结束脚本。应该在您创建的任何自定义驱动程序中注释掉 disable-rpc.fin script。其他更改则是可选的。这些脚本根据其是必需的还是可选的进行了标记。
 - enable-ipfilter.fin - 不启用 IP 过滤器（网络包过滤器）。（可选）
 - disable-rpc.fin - 使远程过程调用 (Remote Procedure Call, RPC) 服务处于启用状态。网络信息服务 (Network Information Service, NIS) 和网络文件系统 (Network File System, NFS) 等许多其他系统服务都使用 RPC 服务。（必需）
 - disable-sma.fin - 使系统管理代理 (NET-SNMP) 处于启用状态。（可选）
 - disable-ssh-root-login.fin - 不能禁用 ssh 超级用户登录。
 - set-term-type.fin - 不需要的旧脚本。（可选）

最小化逻辑域

Solaris OS 可以配有不同数量的软件包，具体取决于您的需求。最小化可以将这组软件包的数量减少到运行所需应用程序所必需的最低数量。最小化非常重要，这是因为它可以减少包含潜在安全漏洞的软件的数量，还可以降低与正确修补已安装软件有关的工作强度。逻辑域最小化活动提供了 JumpStart™ 支持，以便安装仍然完全支持任何域的最小化 Solaris OS。

Solaris Security Toolkit 提供了 JumpStart 配置文件 (minimal-ldm_control.profile)，以便最小化 LDoms 的逻辑域，该配置文件会安装支持 LDoms 和 LDoms MIB 所必需的全部 Solaris OS 软件包。如果希望在控制域上使用 LDoms MIB，则需要在安装 LDoms 和 Solaris Security Toolkit 软件包之后单独添加 LDoms MIB 软件包。LDoms MIB 软件包不会自动与其他软件一起安装。有关安装和使用 LDoms MIB 的更多信息，请参阅《Logical Domains (LDoms) MIB 1.0.1 管理指南》。

授权

Logical Domains Manager 的授权有两个级别：

- 读 – 允许查看，但不能修改配置。
- 读写 – 允许查看和更改配置。

安装 Logical Domains Manager 后，这些更改不会应用到 Solaris OS 中，而是由软件包脚本 postinstall 将其添加到授权文件中。同样，授权条目也是由软件包脚本 preremove 来删除。

下表列出了 ldm 子命令以及执行这些命令所需的相应用户授权。

表 2-1 ldm 子命令及用户授权

ldm 子命令*	用户授权
add-*	solaris.ldoms.write
bind-domain	solaris.ldoms.write
list	solaris.ldoms.read
list-*	solaris.ldoms.read
panic-domain	solaris.ldoms.write
remove-*	solaris.ldoms.write
set-*	solaris.ldoms.write

表 2-1 ldm 子命令及用户授权（续）

ldm 子命令*	用户授权
start-domain	solaris.ldoms.write
stop-domain	solaris.ldoms.write
unbind-domain	solaris.ldoms.write

* 指代您可以添加、列出、删除或设置的所有资源。

审计

审计 Logical Domains Manager CLI 命令是使用 Solaris OS 基本安全模块 (Basic Security Module, BSM) 审计功能来完成的。有关使用 Solaris OS BSM 审计功能的详细信息，请参阅 Solaris 10 的《系统管理指南：安全性服务》。

默认情况下不会为 Logical Domains Manager 启用 BSM 审计功能；但是会提供基础结构。您可以使用以下两种方法之一启用 BSM 审计功能：

- 运行 Solaris Security Toolkit 中的 `enable-bsm.fin` 结束脚本。
- 使用 Solaris OS `bsmconv(1M)` 命令。

有关在 Logical Domains Manager 中使用 BSM 审计功能来启用、检验、禁用、列显输出和轮转日志的详细信息，请参见第 109 页中的“启用并使用 BSM 审计”。

遵从性

Solaris Security Toolkit 具有自己的审计功能。Solaris Security Toolkit 软件通过将运行 Solaris OS 的任何系统的安全状态与预定义的安全配置文件相比较，可以自动验证该系统的安全状态。有关此遵从性功能的更多信息，请参阅《Solaris Security Toolkit 4.2 管理指南》中的“审计系统安全性”。

第3章

安装和启用软件

本章介绍如何在支持的服务器的控制域中安装和启用 Logical Domains Manager 1.0.3 软件及其他软件。有关支持的服务器的列表，请参阅《Logical Domains (LDoms) 1.0.3 发行说明》中的“支持的服务器”。

可以根据您的平台使用本章的所需内容。如果您是在新的 Sun UltraSPARC T2 平台上使用 Logical Domains 软件，则所有软件应该在出厂时进行了预安装。

升级 Solaris OS

本节介绍了有关保存和恢复 Logical Domains 约束数据库文件或对控制域进行即时升级时应了解的信息。

保存和恢复 Logical Domains 约束数据库文件

只要对控制域升级操作系统，您就必须保存和恢复 Logical Domains 约束数据库文件，该文件位于 `/var/opt/SUNWldm/ldom-db.xml` 中。

注 – 在执行会破坏控制域的文件数据（如磁盘交换区）的任何其他操作时，也必须保存和恢复 `/var/opt/SUNWldm/ldom-db.xml` 文件。

对控制域使用即时升级

如果要对控制域使用即时升级，请考虑向 `/etc/lu/synclist` 文件添加以下行：

```
/var/opt/SUNWldm/ldom-db.xml      OVERWRITE
```

这样，在切换引导环境时即可将数据库从活动引导环境自动复制到新的引导环境。有关 `/etc/lu/synclist` 以及在引导环境之间同步文件的更多信息，请参阅《Solaris 10 8/07 安装指南：Solaris Live Upgrade 和升级规划》中的“在引导环境之间同步文件”。

升级到 LDoms 1.0.3 软件

现有的 LDoms 1.0.1 和 1.0.2 配置可用于 LDoms 1.0.3 软件，因此在从 LDoms 1.0.1 或 1.0.2 软件升级到 LDoms 1.0.3 软件时，您不需要执行以下过程。但是，如果要在 LDoms 1.0.3 软件中使用现有的 LDoms 1.0 配置，则需要使用以下过程。

▼ 从 LDoms 1.0 升级到 LDoms 1.0.3 软件

现有的 LDoms 1.0 配置不能用于 LDoms 1.0.3 软件。以下过程介绍使用 XML 约束文件和带有 `-i` 选项的 `ldm start-domain` 命令保存并重建配置的方法。此方法不保留实际的绑定，只保留用于创建那些绑定的约束。这意味着，在完成该过程后，各域会具有相同的虚拟资源，但不一定绑定到相同的物理资源。

此基本过程是要将每个域的约束信息保存到 XML 文件中，这样就可以在升级后向 Logical Domains Manager 重新发出这些约束信息以便重建所需配置。此过程适用于来宾域，而不适用于控制域。虽然可以将控制 (primary) 域的约束保存到 XML 文件，但不能反过来将它提供给 `ldm start-domain -i` 命令。

1. 更新到最新版本的 Solaris OS。有关更多信息，请参见第 16 页中的“安装 Solaris 10 OS”的步骤 2。
2. 为每个域创建一个包含该域约束的 XML 文件。

```
# ldm ls-constraints -x ldom > ldom.xml
```

3. 列出所有存储在系统控制器上的逻辑域配置。

```
# ldm ls-config
```

4. 删除每个存储在系统控制器上的逻辑域配置。

```
# ldm rm-config config_name
```

5. 禁用 Logical Domains Manager 守护进程 (ldmd)。

```
# svcadm disable ldmd
```

6. 删除 Logical Domains Manager 软件包 (SUNWldm)。

```
# pkgrm SUNWldm
```

7. 如果您正在使用 Solaris Security Toolkit 软件包 (SUNWjass)，请删除它。

```
# pkgrm SUNWjass
```

8. 对系统固件进行 Flash 更新。有关整个过程的信息，请参见第 17 页中的“升级系统固件”或第 18 页中的“在不使用 FTP 服务器的情况下升级系统固件”。

9. 下载 LDoms 1.0.3 软件包。

有关下载并安装 Logical Domains Manager、Solaris Security Toolkit 和 Logical Domains MIB 的过程，请参见第 19 页中的“下载 Logical Domains Manager、Solaris Security Toolkit 和 Logical Domains MIB”。

10. 手动重新配置 primary 域。有关说明，请参见第 40 页中的“设置控制域”。

11. 对您在步骤 2 中创建的每个来宾域 XML 文件运行下列命令。

```
# ldm create -i ldom.xml
# ldm bind-domain ldom
# ldm start-domain ldom
```

在控制域上首次安装软件

安装 Logical Domains Manager 软件时创建的第一个域是控制域。该域名为 primary，此名称不可更改。在控制域上将安装下列主要组件。

- Solaris 10 OS。如有必要，请添加《Logical Domains (LDoms) 1.0.3 发行说明》中推荐的任何修补程序。请参见第 16 页中的“安装 Solaris 10 OS”。
- 适用于 Sun UltraSPARC T1 平台的系统固件版本 6.5，或适用于 Sun UltraSPARC T2 平台的系统固件版本 7.0。请参见第 17 页中的“升级系统固件”。

- Logical Domains Manager 1.0.3 软件。请参见第 20 页中的“安装 Logical Domains Manager 和 Solaris Security Toolkit”。
- （可选）Solaris Security Toolkit 4.2 软件。请参见第 20 页中的“安装 Logical Domains Manager 和 Solaris Security Toolkit”。
- （可选）Logical Domains (LDoms) Management Information Base (MIB) 软件包。有关安装和使用 LDoms MIB 的更多信息，请参阅《Logical Domains (LDoms) Management Information Base (MIB) 1.0.1 管理指南》。

必须先在服务器上安装 Solaris OS 和系统固件，然后再安装 Logical Domains Manager。安装完 Solaris OS、系统固件和 Logical Domains Manager 后，最初的域成为控制域。

▼ 安装 Solaris 10 OS

如果尚未安装 Solaris 10 OS，请安装它。要确定应该用于此版本的 Logical Domains 软件的 Solaris 10 OS，请参阅《Logical Domains (LDoms) 1.0.3 发行说明》中的“必需的和推荐的软件”。有关安装 Solaris OS 的完整说明，请参阅 Solaris 10 OS 安装指南。可以根据系统需要调整安装。

注 – 在逻辑域中，只能将 Solaris OS 安装到整个磁盘或作为块设备导出的某个文件中。

1. 安装 Solaris 10 OS。

最小安装是可选的。Solaris Security Toolkit 包含以下适用于 Logical Domains 软件的 JumpStart 最小安装配置文件：

```
/opt/SUNWjass/Profiles/minimal-ldm_control.profile
```

2. 如果正在安装 Solaris 10 11/06 OS，请安装必需的修补程序。有关必需的修补程序的列表，请参阅《Logical Domains (LDoms) 1.0.3 发行说明》中的“必需的 Solaris 10 11/06 OS 修补程序”。

注 – 如果要在来宾域中安装非英语语言的操作系统，控制台终端必须采用 OS 安装程序所需的语言环境。例如，Solaris OS 安装程序需要 EUC 语言环境，而 Linux 安装程序则可能需要 Unicode 语言环境。

▼ 升级系统固件

可以在以下 SunSolve 站点找到适用于您的平台的系统固件：

<http://sunsolve.sun.com>

有关支持的服务器所必需的系统固件的信息，请参阅《Logical Domains (LDoms) 1.0.3 发行说明》中的“必需的系统固件修补程序”。

以下过程介绍如何在系统控制器上使用 `flashupdate(1M)` 命令升级系统固件。

- 如果您无权访问本地 FTP 服务器，请参见第 18 页中的“在不使用 FTP 服务器的情况下升级系统固件”。
- 如果您希望从控制域更新系统固件，请参阅系统固件发行说明。

有关为支持的服务器安装和更新系统固件的更多信息，请参阅这些服务器的管理指南或产品说明。

1. 从连接到系统控制器的任一管理端口（串行端口或网络端口）关闭主机服务器，并关闭其电源。

```
# shutdown -i5 -g0 -y
```

2. 使用 `flashupdate(1M)` 命令升级系统固件，这取决于您的服务器。

```
sc> flashupdate -s IP-address -f path/Sun_System_Firmware-  
x_x_x_build_nn-server-name.bin  
username: your-userid  
password: your-password
```

其中：

- `IP-address` 是您的 FTP 服务器的 IP 地址。
- `path` 是 SunSolvesm 或您自己的目录中可以获取系统固件映像的位置。
- `x_x_x` 是系统固件的版本号。
- `nn` 是适用于此发行版的内部版本号。
- `server-name` 是服务器名称。例如，Sun Fire T2000 服务器的 `server-name` 是 `Sun_Fire_T2000`。

3. 使系统控制器复位。

```
sc> resetsc -y
```

4. 打开主机服务器的电源并引导主机服务器。

```
sc> poweron -c  
ok boot disk
```

▼ 在不使用 FTP 服务器的情况下升级系统固件

如果您无权访问本地 FTP 服务器，无法将固件上传到系统控制器，可以使用 `sysfwdownload` 实用程序，该实用程序随可在以下 SunSolve 站点下载的系统固件升级软件包一起提供：

<http://sunsolve.sun.com>

1. 在 Solaris OS 中运行以下命令。

```
# cd firmware_location  
# sysfwdownload system_firmware_file
```

2. 关闭 Solaris OS 实例。

```
# shutdown -i5 -g0 -y
```

3. 关闭系统控制器的电源，并更新系统控制器上的固件。

```
sc> poweroff -fy  
sc> flashupdate -s 127.0.0.1
```

4. 使系统控制器复位，并打开系统控制器的电源。

```
sc> resetsc -y  
sc> poweron
```

▼ 降级系统固件

在为了使用 Logical Domains 软件而升级系统固件后，您可以将该固件降级到原来的非 Logical Domains 固件。

- 运行 `flashupdate(1M)` 命令，并指定原来的非 Logical Domains 固件的路径。

下载 Logical Domains Manager 和 Solaris Security Toolkit

▼ 下载 Logical Domains Manager、Solaris Security Toolkit 和 Logical Domains MIB

1. 从 Sun 软件下载站点下载包含 **Logical Domains Manager** 软件包 (SUNWldm)、**Solaris Security Toolkit** (SUNWjass) 和安装脚本 (install-ldm) 以及 **Logical Domains Management Information Base** 软件包 (SUNWldmib.v) 的 tar 文件 (LDoms_Manager-1_0_3-04.zip)。可以从以下 Web 站点找到该软件：

<http://www.sun.com/ldoms>

2. 解压缩 zip 文件。

```
$ unzip LDoms_Manager-1_0_3-04.zip
```

下载软件的目录结构类似于下例：

代码示例 3-1 下载的 Logical Domains 1.0.3 软件的目录结构

```
LDoms_Manager-1_0_3/  
  Install/  
    install-ldm  
  Legal/  
    Ldoms_1.0.3_Entitlement.txt  
    Ldoms_1.0.3_SLA_Entitlement.txt  
  Product/  
    SUNWjass/  
    SUNWldm.v/  
    SUNWldmib.v  
  README
```

安装 Logical Domains Manager 和 Solaris Security Toolkit

安装 Logical Domains Manager 和 Solaris Security Toolkit 软件有以下三种方法：

- 使用安装脚本安装软件包和修补程序。此方法将自动安装 Logical Domains Manager 和 Solaris Security Toolkit 这两个软件。请参见第 20 页中的“使用安装脚本安装 Logical Domains Manager 1.0.3 和 Solaris Security Toolkit 4.2 软件”。
- 使用 JumpStart 安装软件包。请参见第 26 页中的“使用 JumpStart 安装 Logical Domains Manager 1.0.3 和 Solaris Security Toolkit 4.2 软件”。
- 手动安装每个软件包。请参见第 29 页中的“手动安装 Logical Domains Manager 和 Solaris Security Toolkit 软件”。

注 – 请切记，在安装完 LDoms 和 Solaris Security Toolkit 软件包后，需要手动安装 LDoms MIB 软件包。LDoms MIB 软件包不会自动与其他软件包一起安装。有关安装和使用 LDoms MIB 的更多信息，请参阅《Logical Domains (LDoms) Management Information Base 1.0.1 管理指南》。

使用安装脚本安装 Logical Domains Manager 1.0.3 和 Solaris Security Toolkit 4.2 软件

如果您使用 `install-ldm` 安装脚本，则可以有几项选择来指定脚本的运行方式。每种选择都在下面的过程中进行了说明。

- 使用不带任何选项的 `install-ldm` 脚本会自动进行以下操作：
 - 检查 Solaris OS 发行版是否为 Solaris 10 11/06
 - 检验软件包子目录 `SUNWldm/` 和 `SUNWjass/` 是否存在
 - 检验必备的 Solaris Logical Domains 驱动程序软件包 `SUNWldomr` 和 `SUNWldomu` 是否存在
 - 检验是否未安装 `SUNWldm` 和 `SUNWjass` 软件包

注 – 如果脚本在安装期间确实检测到以前版本的 `SUNWjass`，则需要删除该版本，但无需撤消先前对 Solaris OS 所进行的任何强化。

- 安装 Logical Domains Manager 1.0.3 软件（SUNWldm 软件包）
- 安装包括必需的修补程序在内的 Solaris Security Toolkit 4.2 软件（SUNWjass 软件包）
- 检验已安装所有软件包
- 启用 Logical Domains Manager 守护进程 ldmd
- 使用 Solaris Security Toolkit `ldm_control-secure.driver` 或以所选的 `-secure.driver` 结尾的其他驱动程序之一，在控制域上强化 Solaris OS。
- 使用带有 `-d` 选项的 `install-ldm` 脚本，允许您指定一个以 `-secure.driver` 结尾的驱动程序以外的 Solaris Security Toolkit 驱动程序。这种选择会自动执行上述选择中列出的所有功能，另外还可以：
 - 使用您指定的 Solaris Security Toolkit 自定义驱动程序（例如，`server-secure-myname.driver`）在控制域上强化 Solaris OS。
- 使用带有 `-d` 选项的 `install-ldm` 脚本并指定 `none`，指定您不想使用 Solaris Security Toolkit 强化运行在控制域上的 Solaris OS。这种选择会自动执行除上述选择中所列的强化以外的所有功能。建议不要跳过 Solaris Security Toolkit 的使用，而且，只有在您打算使用其他过程强化控制域时，才能跳过 Solaris Security Toolkit 的使用。
- 使用带有 `-p` 选项的 `install-ldm` 脚本指定您只想执行启用 Logical Domains Manager 守护进程 (ldmd) 和运行 Solaris Security Toolkit 的安装后操作。例如，如果服务器上预安装了 SUNWldm 和 SUNWjass 软件包，您就可以使用这种选项。请参见第 26 页中的“使用带有 `-p` 选项的 `install-ldm` 脚本进行安装”。

▼ 使用不带任何选项的 `install-ldm` 脚本进行安装

- 运行不带任何选项的安装脚本。

安装脚本是 SUNWldm 软件包的一部分，并位于 `Install` 子目录中。

```
# Install/install-ldm
```

- a. 如果先前安装了一个或多个软件包，则会显示以下消息。

```
# Install/install-ldm
ERROR: One or more packages are already installed: SUNWldm SUNWjass.
If packages SUNWldm.v and SUNWjass are factory pre-installed, run
install-ldm -p to perform post-install actions. Otherwise remove the
package(s) and restart install-ldm.
```

如果您只想执行安装后操作，请转到第 26 页中的“使用带有 `-p` 选项的 `install-ldm` 脚本进行安装”。

b. 如果该过程成功，则会显示类似于以下示例的消息。

- 代码示例 3-2 显示了在选择以下默认安全配置文件的情况下成功运行 `install-ldm` 脚本的输出：
 - a) Hardened Solaris configuration for LDoms (recommended)
- 代码示例 3-3 显示了在选择以下安全配置文件的情况下成功运行 `install-ldm` 脚本的输出：
 - c) Your custom-defined Solaris security configuration profile

所显示的供您选择的驱动程序是以 `-secure.driver` 结尾的驱动程序。如果您编写的自定义驱动程序不是以 `-secure.driver` 结尾，则必须使用 `install-ldm -d` 选项指定您的自定义驱动程序。（请参见第 24 页中的“使用带有 `-d` 选项的 `install-ldm` 脚本进行安装”。）

代码示例 3-2 LDoms 强化 Solaris 配置的输出

```
# Install/install-ldm
Welcome to the LDoms installer.

You are about to install the domain manager package that will enable
you to create, destroy and control other domains on your system. Given
the capabilities of the domain manager, you can now change the security
configuration of this Solaris instance using the Solaris Security
Toolkit.

Select a security profile from this list:

a) Hardened Solaris configuration for LDoms (recommended)
b) Standard Solaris configuration
c) Your custom-defined Solaris security configuration profile

Enter a, b, or c [a]: a
The changes made by selecting this option can be undone through the
Solaris Security Toolkit's undo feature. This can be done with the
'/opt/SUNWjass/bin/jass-execute -u' command.

Installing LDoms and Solaris Security Toolkit packages.
pkgadd -n -d "/var/tmp/install/Product/Logical_Domain_Manager" -a pkg_admin
SUNWldm.v
Copyright 2006 Sun Microsystems, Inc. All rights reserved.
Use is subject to license terms.

Installation of <SUNWldm> was successful.
pkgadd -n -d "/var/tmp/install/Product/Solaris_Security_Toolkit" -a pkg_admin
SUNWjass
Copyright 2005 Sun Microsystems, Inc. All rights reserved.
Use is subject to license terms.
```

```
Installation of <SUNWjass> was successful.

Verifying that all packages are fully installed.  OK.
Enabling services: svc:/ldoms/ldmd:default
Running Solaris Security Toolkit 4.2.0 driver ldm_control-secure.driver.
Please wait. . .
/opt/SUNWjass/bin/jass-execute -q -d ldm_control-secure.driver
Executing driver, ldm_control-secure.driver
Solaris Security Toolkit hardening executed successfully; log file
/var/opt/SUNWjass/run/20070208142843/jass-install-log.txt.  It will not
take effect until the next reboot.  Before rebooting, make sure SSH or
the serial line is setup for use after the reboot.
```

```
# Install/install-ldm
Welcome to the LDoms installer.

You are about to install the domain manager package that will enable
you to create, destroy and control other domains on your system. Given
the capabilities of the domain manager, you can now change the security
configuration of this Solaris instance using the Solaris Security
Toolkit.

Select a security profile from this list:

a) Hardened Solaris configuration for LDoms (recommended)
b) Standard Solaris configuration
c) Your custom-defined Solaris security configuration profile

Enter a, b, or c [a]: c
Choose a Solaris Security Toolkit .driver configuration profile from
this list
1) ldm_control-secure.driver
2) secure.driver
3) server-secure.driver
4) suncluster3x-secure.driver
5) sunfire_15k_sc-secure.driver

Enter a number 1 to 5: 2
The driver you selected may not perform all the LDoms-specific
operations specified in the LDoms Administration Guide.
Is this OK (yes/no)? [no] y
The changes made by selecting this option can be undone through the
Solaris Security Toolkit's undo feature. This can be done with the
'/opt/SUNWjass/bin/jass-execute -u' command.
```

```
Installing LDoms and Solaris Security Toolkit packages.
pkgadd -n -d "/var/tmp/install/Product/Logical_Domain_Manager" -a pkg_admin
SUNWldm.v
Copyright 2006 Sun Microsystems, Inc. All rights reserved.
Use is subject to license terms.

Installation of <SUNWldm> was successful.
pkgadd -n -d "/var/tmp/install/Product/Solaris_Security_Toolkit" -a pkg_admin
SUNWjass
Copyright 2005 Sun Microsystems, Inc. All rights reserved.
Use is subject to license terms.

Installation of <SUNWjass> was successful.

Verifying that all packages are fully installed. OK.
Enabling services: svc:/ldoms/ldmd:default
Running Solaris Security Toolkit 4.2.0 driver secure.driver.
Please wait. . .
/opt/SUNWjass/bin/jass-execute -q -d secure.driver
Executing driver, secure.driver
Solaris Security Toolkit hardening executed successfully; log file
/var/opt/SUNWjass/run/20070102142843/jass-install-log.txt. It will not
take effect until the next reboot. Before rebooting, make sure SSH or
the serial line is setup for use after the reboot.
```

▼ 使用带有 -d 选项的 install-ldm 脚本进行安装

- 运行带有 -d 选项的安装脚本，以指定 **Solaris Security Toolkit** 自定义强化驱动程序，例如 `server-secure-myname.driver`。
安装脚本是 SUNWldm 软件包的一部分，并位于 Install 子目录中。

```
# Install/install-ldm -d server-secure-myname.driver
```

如果该过程成功，则会显示类似于代码示例 3-4 中的消息。

```
# Install/install-ldm -d server-secure.driver
The driver you selected may not perform all the LDoms-specific
operations specified in the LDoms Administration Guide.
Installing LDoms and Solaris Security Toolkit packages.
pkgadd -n -d "/var/tmp/install/Product/Logical_Domain_Manager" -a pkg_admin
SUNWldm.v
Copyright 2006 Sun Microsystems, Inc. All rights reserved.
Use is subject to license terms.
```

代码示例 3-4 install-ldm -d 脚本运行成功时的输出（续）

```
Installation of <SUNWldm> was successful.
pkgadd -n -d "/var/tmp/install/Product/Solaris_Security_Toolkit" -a pkg_admin
SUNWjass
Copyright 2005 Sun Microsystems, Inc. All rights reserved.
Use is subject to license terms.

Installation of <SUNWjass> was successful.

Verifying that all packages are fully installed. OK.
Enabling services: svc:/ldoms/ldmd:default
Running Solaris Security Toolkit 4.2.0 driver server-secure-myname.driver.
Please wait. . .
/opt/SUNWjass/bin/jass-execute -q -d server-secure-myname.driver
Executing driver, server-secure-myname.driver
Solaris Security Toolkit hardening executed successfully; log file
/var/opt/SUNWjass/run/20061114143128/jass-install-log.txt. It will not
take effect until the next reboot. Before rebooting, make sure SSH or
the serial line is setup for use after the reboot.
```

▼ 使用带有 -d none 选项的 install-ldm 脚本进行安装

- 运行带有 -d none 选项的安装脚本，以指定不使用 Solaris Security Toolkit 驱动程序强化您的系统。

安装脚本是 SUNWldm 软件包的一部分，并位于 Install 子目录中。

```
# Install/install-ldm -d none
```

如果该过程成功，则会显示类似于代码示例 3-5 中所示的消息。

代码示例 3-5 install-ldm -d none 脚本运行成功时的输出

```
# Install/install-ldm -d none
Installing LDomS and Solaris Security Toolkit packages.
pkgadd -n -d "/var/tmp/install/Product/Logical_Domain_Manager" -a pkg_admin
SUNWldm.v
Copyright 2006 Sun Microsystems, Inc. All rights reserved.
Use is subject to license terms.

Installation of <SUNWldm> was successful.
pkgadd -n -d "/var/tmp/install/Product/Solaris_Security_Toolkit" -a pkg_admin
SUNWjass
Copyright 2005 Sun Microsystems, Inc. All rights reserved.
Use is subject to license terms.
```

```
Installation of <SUNWjass> was successful.

Verifying that all packages are fully installed.  OK.
Enabling services: svc:/ldoms/ldmd:default
Solaris Security Toolkit was not applied. Bypassing the use of the
Solaris Security Toolkit is not recommended and should only be
performed when alternative hardening steps are to be taken.
```

▼ 使用带有 -p 选项的 install-ldm 脚本进行安装

如果 SUNWldm 和 SUNWjass 软件包已预安装在服务器上，并且您想执行启用 Logical Domains Manager 守护进程 (ldmd) 和运行 Solaris Security Toolkit 的安装后操作，则可以使用这种选择。

- 运行带有 -p 选项的安装脚本，只执行启用 ldmd 和运行 Solaris Security Toolkit 的安装后操作以强化您的系统。

```
# Install/install-ldm -p
Verifying that all packages are fully installed.  OK.
Enabling services: svc:/ldoms/ldmd:default
Running Solaris Security Toolkit 4.2.0 driver ldm_control-secure.driver.
Please wait. . .
/opt/SUNWjass/bin/jass-execute -q -d ldm_control-secure.driver
Solaris Security Toolkit hardening executed successfully; log file
var/opt/SUNWjass/run/20070515140944/jass-install-log.txt.  It will not
take effect until the next reboot.  Before rebooting, make sure SSH or
the serial line is setup for use after the reboot.
```

使用 JumpStart 安装 Logical Domains Manager 1.0.3 和 Solaris Security Toolkit 4.2 软件

有关使用 JumpStart 的完整信息，请参阅《JumpStart Technology: Effective Use in the Solaris Operating Environment》。



注意 – 在网络安装期间，不要与虚拟控制台断开连接。

▼ 安装 JumpStart 服务器

- 如果您已经安装了 JumpStart 服务器，请继续本管理指南的 [第 27 页](#) 中的“使用 JumpStart 软件进行安装”。
- 如果您尚未安装 JumpStart 服务器，则必须执行该操作。

有关此过程的完整信息，请参阅《Solaris 10 11/06 安装指南：自定义 JumpStart 和高级安装》。可以在以下网址找到此安装指南：

<http://docs.sun.com/app/docs/doc/819-7825>

1. 参阅《Solaris 10 11/06 安装指南：自定义 JumpStart 和高级安装》中的第 3 章“准备自定义 JumpStart 安装（任务）”，并执行下列步骤。
 - a. 阅读“任务图：准备自定义 JumpStart 安装”中的任务图。
 - b. 使用“为联网系统创建配置文件服务器”中的过程设置联网系统。
 - c. 使用“创建 rules 文件”中的过程创建 rules 文件。
2. 使用“验证 rules 文件”中的过程验证 rules 文件。

Solaris Security Toolkit 提供了配置文件和结束脚本。有关配置文件和结束脚本的更多信息，请参阅《Solaris Security Toolkit 4.2 Reference Manual》。

▼ 使用 JumpStart 软件进行安装

1. 转到您下载 Solaris Security Toolkit 软件包 (SUNWjass) 的目录。

```
# cd /path-to-download
```

2. 安装 SUNWjass 以便创建 JumpStart (jumpstart) 目录结构。

```
# pkgadd -R /jumpstart -d . SUNWjass
```

3. 使用文本编辑器修改
/jumpstart/opt/SUNWjass/Sysidcfg/Solaris_10/sysidcfg 文件以反映您的网络环境。
4. 将 /jumpstart/opt/SUNWjass/Drivers/user.init.SAMPLE 文件复制到
/jumpstart/opt/SUNWjass/Drivers/user.init 文件。

```
# cp user.init.SAMPLE user.init
```

5. 编辑 user.init 文件以反映您的路径。

6. 要在 **JumpStart** 安装期间将 **Solaris Security Toolkit** 软件包 (SUNWjass) 安装到目标系统上, 必须将该软件包放到 `user.init` 文件中定义的 `JASS_PACKAGE_MOUNT` 目录中。例如:

```
# cp -r /path/to/LDoms_Manager-1_0_2/Product/SUNWjass
/jumpstart/opt/SUNWjass/Packages
```

7. 要在 **JumpStart** 安装期间将 **Logical Domains Manager** 软件包 (SUNWldm.v) 安装到目标系统上, 必须将该软件包从下载区移到 `user.init` 文件中定义的 `JASS_PACKAGE_MOUNT` 目录中。例如:

```
# cp -r /path/to/LDoms_Manager-1_0_2/Product/SUNWldm.v
/jumpstart/opt/SUNWjass/Packages
```

8. 如果在使用多宿主 **JumpStart** 服务器时遇到问题, 请将 `user.init` 文件中与 `JASS_PACKAGE_MOUNT` 和 `JASS_PATCH_MOUNT` 对应的两项修改为指向目录 `JASS_HOME_DIR/Patches` 和 `JASS_HOME_DIR/Packages` 的正确路径。有关更多信息, 请参阅 `user.init.SAMPLE` 文件中的注释。
9. 将 `ldm_control-secure.driver` 用作 **Logical Domains Manager** 控制域的基本驱动程序。
- 有关如何按您自己的使用需要修改该驱动程序的信息, 请参阅《Solaris Security Toolkit 4.2 Reference Manual》的第 4 章。Solaris Security Toolkit 中与 `ldm_control-secure.driver` 对应的主驱动程序是 `secure.driver`。
10. 完成对 `ldm_control-secure.driver` 的修改后, 在 `rules` 文件中创建正确的条目。
- 如果想最小化 LDoms 控制域, 请在您的 `rules` 文件中按如下方式指定 `minimal-ldm-control.profile`。

```
hostname imbulu - Profiles/minimal-ldm_control.profile
Drivers/ldm_control-secure-abc.driver
```

注 – 请切记, 在安装完 LDoms 和 Solaris Security Toolkit 软件包后, 需要手动安装 LDoms MIB 软件包。LDoms MIB 软件包不会自动与其他软件包一起安装。有关安装和使用 LDoms MIB 的更多信息, 请参阅《Logical Domains (LDoms) Management Information Base 1.0.1 管理指南》。

- 如果不想最小化 LDoms 控制域, 请按如下方式输入命令。

```
hostname imbulu - Profiles/oem.profile Drivers/ldm_control-secure-abc.driver
```

11. 如果在 JumpStart 安装期间撤消了强化，则必须运行以下 SMF 命令重新启动 Logical Domains Manager。

```
# svcadm enable svc:/ldoms/ldmd:default
```

手动安装 Logical Domains Manager 和 Solaris Security Toolkit 软件

请执行下列过程，手动安装 Logical Domains Manager 和 Solaris Security Toolkit 软件：

- 第 29 页中的 “手动安装 Logical Domains Manager (LDoms) 1.0.3 软件”
- 第 30 页中的 “(可选) 手动安装 Solaris Security Toolkit 4.2 软件”
- 第 30 页中的 “(可选) 手动强化控制域”

▼ 手动安装 Logical Domains Manager (LDoms) 1.0.3 软件

从 Sun 软件下载站点下载 Logical Domains Manager 1.0.3 软件，即 SUNWldm 软件包。有关具体说明，请参见第 19 页中的 “下载 Logical Domains Manager、Solaris Security Toolkit 和 Logical Domains MIB”。

1. 使用 pkgadd(1M) 命令安装 SUNWldm.v 软件包。使用 -G 选项仅在全局区域中安装该软件包，使用 -d 选项指定 SUNWldm.v 软件包所在目录的路径。

```
# pkgadd -Gd . SUNWldm.v
```

2. 对交互式提示中的所有问题回答 y（代表“是”）。
3. 使用 pkginfo(1) 命令检验是否安装了 Logical Domains Manager 1.0.3 软件包的 SUNWldm 软件包。

下面显示的修订 (REV) 信息是一个示例。

```
# pkginfo -l SUNWldm | grep VERSION
VERSION=1.0.3,REV=2007.08.23.10.20
```

▼ （可选）手动安装 Solaris Security Toolkit 4.2 软件

如果想保护系统的安全，请下载并安装 SUNWjass 软件包。必需的修补程序（122608-03 和 125672-01）包括在 SUNWjass 软件包中。有关下载该软件的具体说明，请参见第 19 页中的“[下载 Logical Domains Manager、Solaris Security Toolkit 和 Logical Domains MIB](#)”。

有关使用 Logical Domains Manager 软件时的安全注意事项的更多信息，请参见本文档的第 2 章。如需进一步参考，可以在以下网址查找 Solaris Security Toolkit 4.2 文档：

<http://docs.sun.com>

1. 使用 `pkgadd(1M)` 命令安装 SUNWjass 软件包。

```
# pkgadd -d . SUNWjass
```

2. 使用 `pkginfo(1)` 命令检验是否安装了 Solaris Security Toolkit 4.2 软件包的 SUNWjass 软件包。

```
# pkginfo -l SUNWjass | grep VERSION
VERSION: 4.2.0
```

▼ （可选）手动强化控制域

仅当您安装了 Solaris Security Toolkit 4.2 软件包时，才执行以下过程。

注 – 如果使用 Solaris Security Toolkit 强化控制域，需要禁用很多系统服务并对网络访问施加某些限制。请参阅本书的第 xix 页中的“[相关文档](#)”以查找 Solaris Security Toolkit 4.2 文档，了解更多信息。

1. 使用 `ldm_control-secure.driver` 进行强化。

```
# /opt/SUNWjass/bin/jass-execute -d ldm_control-secure.driver
```

您可以使用其他驱动程序强化系统，还可以为调整环境安全性而自定义驱动程序。有关驱动程序和自定义驱动程序的更多信息，请参阅《Solaris Security Toolkit 4.2 Reference Manual》。

2. 对交互式提示中的所有问题回答 `y`（代表“是”）。
3. 关闭并重新引导服务器，使强化生效。

```
# /usr/sbin/shutdown -y -g0 -i6
```

▼ 验证强化

- 检查 **Logical Domains** 强化驱动程序 (`ldom_control-secure.driver`) 是否正确应用了强化。

如果想检查其他驱动程序，只需替换此命令示例中的驱动程序名称。

```
# /opt/SUNWjass/bin/jass-execute -a ldom_control-secure.driver
```

▼ 撤消强化

1. 撤消由 **Solaris Security Toolkit** 应用的配置更改。

```
# /opt/SUNWjass/bin/jass-execute -u
```

Solaris Security Toolkit 会询问您要撤消哪些强化操作。

2. 选择您要撤消的强化操作。
3. 重新引导系统，使取消强化的配置生效。

```
# /usr/sbin/shutdown -y -g0 -i6
```

注 – 如果您要撤消在 JumpStart 安装期间执行的强化，则必须运行以下 SMF 命令重新启动 Logical Domains Manager 和虚拟网络终端服务器守护进程。

```
# svcadm enable svc:/ldoms/ldmd:default
```

启用 Logical Domains Manager 守护进程

安装脚本 `install-ldm` 会自动启用 Logical Domains Manager 守护进程 (`ldmd`)。如果已经手动安装了 Logical Domains Manager 软件，则必须启用 Logical Domains Manager 守护进程 `ldmd`，通过该守护进程可以创建、修改和控制逻辑域。

▼ 启用 Logical Domains Manager 守护进程

1. 使用 `svcadm(1M)` 命令启用 **Logical Domains Manager** 守护进程 `ldmd`。

```
# svcadm enable ldmd
```

2. 使用 `ldm list` 命令检验是否在运行 **Logical Domains Manager**。

您会收到类似于以下内容的消息，这是对于 `factory-default` 配置而言的。请注意，`primary` 域处于 `active` 状态，这表示 **Logical Domains Manager** 正在运行。

```
# /opt/SUNWldm/bin/ldm list
```

NAME	STATE	FLAGS	CONS	VCPU	MEMORY	UTIL	UPTIME
primary	active	---c-	SP	32	3264M	0.3%	19d 9m

创建授权和配置文件并为用户帐户分配角色

需要使用适用于 **Logical Domains Manager** 的 Solaris OS 基于角色的访问控制 (**Role-Based Access Control, RBAC**)，来设置授权和配置文件并为用户帐户分配角色。有关 **RBAC** 的更多信息，请参阅 **Solaris 10 System Administrator Collection**。

Logical Domains Manager 的授权有两个级别：

- 读 – 允许查看，但不能修改配置。
- 读写 – 允许查看和更改配置。

以下为自动添加到 Solaris OS `/etc/security/auth_attr` 文件中的 **Logical Domains** 条目：

- `solaris.ldoms:::LDom administration::`
- `solaris.ldoms.grant:::Delegate LDom configuration::`
- `solaris.ldoms.read:::View LDom configuration::`
- `solaris.ldoms.write:::Manage LDom configuration::`

管理用户授权

▼ 为用户添加授权

根据需要执行下列步骤，在 `/etc/security/auth_attr` 文件中为 Logical Domains Manager 用户添加授权。由于超级用户已经具有 `solaris.*` 授权，因此超级用户已经具有 `solaris.ldoms.*` 授权的相应权限。

1. 为每个需要授权使用 `ldm(1M)` 子命令的用户创建一个本地用户帐户。

注 – 要为用户添加 Logical Domains Manager 授权，必须为该用户创建一个本地（非 LDAP）帐户。有关详细信息，请参阅 Solaris 10 System Administrator Collection。

2. 执行下列操作之一，具体取决于您希望该用户能够使用哪个 `ldm(1M)` 子命令。

有关 `ldm(1M)` 命令及其用户授权的列表，请参见表 2-1。

- 使用 `usermod(1M)` 命令为用户添加只读授权。

```
# usermod -A solaris.ldoms.read username
```

- 使用 `usermod(1M)` 命令为用户添加读写授权。

```
# usermod -A solaris.ldoms.write username
```

▼ 为用户删除所有授权

- 为本地用户帐户删除所有授权（唯一可能的选择）。

```
# usermod -A `` username
```

管理用户配置文件

SUNWldm 软件包会在 `/etc/security/prof_attr` 文件中添加两个系统定义的 RBAC 配置文件，以便授权非超级用户访问 Logical Domains Manager。这两个 LDoms 特定的配置文件为：

- LDoms Review:::Review LDoms configuration:auths=solaris.ldoms.read
- LDoms Management:::Manage LDoms domains:auths=solaris.ldoms.*

可以使用以下过程将上述其中一个配置文件分配给某个用户帐户。

▼ 为用户添加配置文件

- 为本地用户帐户添加管理配置文件，例如 **LDoms Management**。

```
# usermod -P "LDoms Management" username
```

▼ 为用户删除所有配置文件

- 为本地用户帐户删除所有配置文件（唯一可能的选择）。

```
# usermod -P `` username
```

为用户分配角色

使用此过程的优势是，只有被分配了特定角色的用户才能承担该角色。在承担角色的过程中，如果该角色指定有一个密码，则需要输入该密码。这样便实现了双层安全性。如果没有为用户分配角色，则此用户不能承担该角色（通过执行 `su role_name` 命令），即使此用户有正确的密码也是如此。

▼ 创建角色并将该角色分配给用户

1. 创建角色。

```
# roleadd -A solaris.ldoms.read ldm_read
```

2. 为该角色指定密码。

```
# passwd ldm_read
```

3. 将该角色分配给用户，例如 `user_1`。

```
# useradd -R ldm_read user_1
```

4. 为该用户 (`user_1`) 指定密码。

```
# passwd user_1
```


5. 仅向 `user_1` 帐户分配访问权限，使其成为 `ldm_read` 帐户。

```
# su user_1
```

6. 出现提示时，键入用户密码。
7. 检验用户 **ID** 和 `ldm_read` 角色的访问权限。

```
$ id
uid=nn(user_1) gid=nn(<group name>)
$ roles
ldm_read
```

8. 向用户提供对具有读授权的 `ldm` 子命令的访问权限。

```
# su ldm_read
```

9. 出现提示时，键入用户密码。
10. 键入 `id` 命令以显示该用户。

```
$ id
uid=nn(ldm_read) gid=nn(<group name>)
```


第4章

设置服务和逻辑域

本章介绍了设置默认服务以及控制域和来宾域所必需的过程。

输出消息

您会从用于创建默认服务和用于设置控制 (primary) 域的命令收到不同的输出消息，具体取决于您的平台：

- Sun UltraSPARC T1 处理器
- Sun UltraSPARC T2 处理器

Sun UltraSPARC T1 处理器

如果您使用的是配有 Sun UltraSPARC T1 处理器的服务器，则在执行了 primary 域的设置命令后，会收到以下通知：

```
Notice: the LDom Manager is running in configuration mode. Any
configuration changes made will only take effect after the machine
configuration is downloaded to the system controller and the host
is reset.
```

Sun UltraSPARC T2 处理器

首次操作 – 如果您使用的是配有 Sun UltraSPARC T2 处理器的服务器，则在对任何设备或者对 `primary` 域上的任何服务首次执行操作之后，会收到以下消息：

```
Initiating delayed reconfigure operation on LDom primary. All
configuration changes for other LDom s are disabled until the
LDom reboots, at which time the new configuration for LDom
primary will also take effect.
```

重新引导前的后续操作 – 如果您使用的是配有 Sun UltraSPARC T2 处理器的服务器，则在重新引导之前每次对 `primary` 域执行后续操作之后，会收到以下通知：

```
Notice: LDom primary is in the process of a delayed
reconfiguration. Any changes made to this LDom will only take
effect after it reboots.
```

创建默认服务

最初必须创建下列虚拟默认服务，以后才能使用它们：

- `vdiskserver` – 虚拟磁盘服务器
- `vswitch` – 虚拟交换机服务
- `vconscon` – 虚拟控制台集中器服务

▼ 创建默认服务

1. 创建虚拟磁盘服务器 (`vds`)，以便允许将虚拟磁盘导入到逻辑域中。

例如，以下命令可将虚拟磁盘服务器 (`primary-vds0`) 添加到控制域 (`primary`)。

```
primary$ ldm add-vds primary-vds0 primary
```

2. 创建虚拟控制台集中器服务 (`vcc`)，以便供虚拟网络终端服务器守护进程 (`vntsd`) 使用，并将该服务作为所有逻辑域控制台的集中器。

例如，以下命令会将端口范围从 5000 至 5100 的虚拟控制台集中器服务 (`primary-vcc0`) 添加到控制域 (`primary`)。

```
primary$ ldm add-vcc port-range=5000-5100 primary-vcc0 primary
```

3. 创建虚拟交换机服务 (vsw)，以启用逻辑域中的虚拟网络 (vnet) 设备之间的联网。如果每个逻辑域需要通过虚拟交换机与外界进行通信，请将 GLDv3 兼容的网络适配器分配给虚拟交换机。

例如，以下命令会将网络适配器驱动程序 e1000g0 的虚拟交换机服务 (primary-vsw0) 添加到控制域 (primary)。

```
primary$ ldm add-vsw net-dev=e1000g0 primary-vsw0 primary
```

此命令会自动将 MAC 地址分配到虚拟交换机。您可以将自己的 MAC 地址指定为 ldm add-vsw 命令的一个选项。但在那种情况下，您必须负责保证所指定的 MAC 地址不会与已经存在的 MAC 地址相冲突。

如果添加的虚拟交换机将取代底层物理适配器作为主网络接口，则必须为其分配物理适配器的 MAC 地址，以便动态主机配置协议 (Dynamic Host Configuration Protocol, DHCP) 服务器会为该域分配相同的 IP 地址。请参见第 42 页中的“启用控制/服务域与其他域之间的联网”。

```
primary$ ldm add-vsw mac-addr=2:04:4f:fb:9f:0d net-dev=e1000g0 primary-vsw0 primary
```

注 – 您每次在域重新配置过程中添加虚拟交换机设备时，都一定要针对所做的重新配置进行重新引导。在设置控制域的过程中，通常需要这样做。有关更多信息，请参见第 42 页中的“重新引导以使用逻辑域”。

4. 使用 list-services 子命令检验是否已创建服务。输出应该与以下内容类似。

```
primary$ ldm list-services primary
VDS
  NAME          VOLUME          OPTIONS          DEVICE
  primary-vds0
VCC
  NAME          PORT-RANGE
  primary-vcc0  5000-5100
VSW
  NAME          MAC          NET-DEV          DEVICE          MODE
  primary-vsw0  02:04:4f:fb:9f:0d e1000g0          switch@0        prog,promisc
```

控制域的初始配置

最初，所有系统资源都分配给控制域。要允许创建其他逻辑域，您必须释放其中一些资源。

注 – 在下列示例的输出中，LDoms Manager 在配置模式下运行时出现的通知仅适用于 Sun UltraSPARC T1 处理器。

▼ 设置控制域

注 – 以下过程举例说明了为控制域设置的资源。这些数字仅为示例，并且所使用的值可能不适用于您的控制域。

1. 为控制域分配加密资源。

注 – 如果控制域中存在任何加密设备，则不能动态地重新配置 CPU。因此，如果您没有在使用加密设备，请将 `set-mau` 设置为 0。

以下示例会为控制域 `primary` 分配一个加密资源。这会令其余的加密资源可供来宾域使用。

```
primary$ ldm set-mau 1 primary
```

2. 为控制域分配虚拟 CPU。

例如，以下命令会为控制域 `primary` 分配 4 个虚拟 CPU。这会令其余的虚拟 CPU 可供来宾域使用。

```
primary$ ldm set-vcpu 4 primary
```

3. 为控制域分配内存。

例如，以下命令会为控制域 `primary` 分配 1 千兆字节的内存。这会令其余的内存可供来宾域使用。

```
primary$ ldm set-memory 1G primary
```

注 – 如果不使用 ZFS 提供磁盘服务，1 GB 内存应该已足够。如果使用 ZFS 提供磁盘服务，请分配 4 个虚拟 CPU 的完整内核以及至少 4 GB 内存。可能需要分配更多的完整内核，以便处理较重的 I/O 负载。

4. 将逻辑域机器配置添加到系统控制器 (system controller, SC)。

例如，以下命令会添加名为 `initial` 的配置。

```
primary$ ldm add-config initial
```

注 – 目前，SC 上可以保存的配置数限制为 8 个，不包括 `factory-default` 配置。

5. 检验该配置在下次重新引导时可以使用。

```
primary$ ldm list-config  
factory-default [current]  
initial [next]
```

此 `list` 子命令显示了当前正在使用 `factory-default` 配置集，一旦重新引导，将会使用 `initial` 配置集。

重新引导以使用逻辑域

必须重新引导控制/服务域，才能使配置更改生效，并释放资源供其他逻辑域使用。

▼ 重新引导

1. 在下次引导时启动重新配置。

```
primary# touch /reconfigure
```

注 — 只有当在域重新配置过程中添加了虚拟交换机设备时，才需要执行这个重新配置步骤。在设置控制域的过程中，通常需要这样做。

2. 关闭并重新引导 primary 域，它也是我们示例中的服务域。

```
primary# shutdown -y -g0 -i6
```

注 — 重新引导时，使用指定的命令使所做的更改生效，但 `ldm list-config` 命令显示的输出仍然与重新引导之前显示的输出相同。要使 `ldm list-config` 命令更新所显示的配置，需要关闭电源再打开电源。

启用控制/服务域与其他域之间的联网

默认情况下，会禁用系统中控制/服务域与其他域之间的联网。要启用此功能，应将虚拟交换机设备配置为网络设备。虚拟交换机可以取代底层物理设备（在此示例中为 `e1000g0`）作为主接口，或者配置为域中的附加网络接口。

注 — 请从域控制台执行下列配置步骤，因为该过程可能会暂时中断与域的网络连接。

▼ 将虚拟交换机配置为主接口

1. 列显所有接口的寻址信息。

```
primary# ifconfig -a
```

2. 激活 (**plumb**) 虚拟交换机。在此示例中，`vsw0` 是所配置的虚拟交换机。

```
primary# ifconfig vsw0 plumb
```

3. (可选) 要获得域中所有虚拟交换机实例的列表，您可以列出它们。

```
primary# /usr/sbin/dladm show-link | grep vsw
vsw0                type: non-vlan  mtu: 1500      device: vsw0
```

4. 取消激活 (**unplumb**) 分配给虚拟交换机 (`net-dev`) 的物理网络设备，在此示例中为 `e1000g0`。

```
primary# ifconfig e1000g0 down unplumb
```

5. 要将物理网络设备 (`e1000g0`) 的属性迁移到虚拟交换机 (`vsw0`) 设备，请执行以下操作之一：

- 如果联网是使用静态 IP 地址配置的，请对 `vsw0` 重用 `e1000g0` 的 IP 地址和网络掩码。

```
primary# ifconfig vsw0 IP_of_e1000g0 netmask netmask_of_e1000g0 broadcast + up
```

- 如果联网是使用 DHCP 配置的，请对 `vsw0` 启用 DHCP。

```
primary# ifconfig vsw0 dhcp start
```

6. 进行必需的配置文件修改，使其成为永久性更改。

```
primary# mv /etc/hostname.e1000g0 /etc/hostname.vsw0
primary# mv /etc/dhcp.e1000g0 /etc/dhcp.vsw0
```

注 — 如有必要，也可以配置虚拟交换机和物理网络设备。这种情况下，需要像步骤 2 中一样激活 (**plumb**) 虚拟交换机，但不需要取消激活 (**unplumb**) 物理设备（跳过步骤 4）。随后，必须将该虚拟交换机配置为具有静态 IP 地址，或者从 DHCP 服务器获取动态 IP 地址。

启用虚拟网络终端服务器守护进程

必须启用虚拟网络终端服务器守护进程 (vntsd)，以便提供对每个逻辑域的虚拟控制台的访问。有关如何使用该守护进程的信息，请参阅 Solaris 10 OS Reference Manual collection 或 vntsd(1M) 手册页。

▼ 启用虚拟网络终端服务器守护进程

注 – 请确保在启用 vntsd 之前已经在控制域上创建了默认服务 vconscon。有关更多信息，请参见第 38 页中的“创建默认服务”。

1. 使用 **svcadm(1M)** 命令启用虚拟网络终端服务器守护进程 **vntsd(1M)**。

```
# svcadm enable vntsd
```

2. 使用 **svcs(1)** 命令检验 vntsd 是否已启用。

```
# svcs -l vntsd
fmri          svc:/ldoms/vntsd:default
enabled      true
state         online
next_state    none
state_time    Sat Jan 27 03:14:17 2007
logfile       /var/svc/log/ldoms-vntsd:default.log
restarter     svc:/system/svc/restarter:default
contract_id   93
dependency    optional_all/error svc:/milestone/network (online)
dependency    optional_all/none svc:/system/system-log (online)
```

创建和启动来宾域

来宾域必须运行既可以识别 `sun4v` 平台又可以识别由虚拟机管理程序提供的虚拟设备的操作系统。目前，该操作系统至少为 Solaris 10 11/06 OS。有关可能需要的任何特定修补程序的信息，请参阅《Logical Domains (LDoms) 1.0.3 发行说明》。一旦创建了默认服务并重新分配了控制域的资源，您就可以创建和启动来宾域。

▼ 创建和启动来宾域

1. 创建逻辑域。

例如，以下命令会创建名为 `ldg1` 的来宾域。

```
primary$ ldm add-domain ldg1
```

2. 为来宾域添加 CPU。

例如，以下命令会为来宾域 `ldg1` 添加四个虚拟 CPU。

```
primary$ ldm add-vcpu 4 ldg1
```

3. 为来宾域添加内存。

例如，以下命令会为来宾域 `ldg1` 添加 512 兆字节的内存。

```
primary$ ldm add-memory 512m ldg1
```

4. 为来宾域添加虚拟网络设备。

例如，以下命令会为来宾域 `ldg1` 添加具有下列特定信息的虚拟网络设备。

```
primary$ ldm add-vnet vnet1 primary-vsw0 ldg1
```

其中：

- `vnet1` 是逻辑域的唯一接口名称，它被分配到此虚拟网络设备实例，供在后续 `set-vnet` 或 `remove-vnet` 子命令上引用。
- `primary-vsw0` 是要连接到的现有网络服务（虚拟交换机）的名称。

5. 指定要由虚拟磁盘服务器导出为来宾域的虚拟磁盘的设备。

可以将物理磁盘、磁盘片、卷或文件导出为块设备。下面以物理磁盘和文件为例进行了说明。

- **物理磁盘示例。**第一个示例会添加具有下列特定信息的物理磁盘。

```
primary$ ldm add-vdsdev /dev/dsk/c0t0d0s2 voll@primary-vds0
```

其中：

- */dev/dsk/c0t0d0s2* 是实际物理设备的路径名。添加设备时，路径名必须与设备名称成对出现。
- *voll* 是必须为添加到虚拟磁盘服务器的设备指定的唯一名称。该卷名称对于此虚拟磁盘服务器实例必须是唯一的，这是因为该名称将由此虚拟磁盘服务器导出到客户机以便进行添加。添加设备时，卷名称必须与实际设备的路径名成对出现。
- *primary-vds0* 是要将此设备添加到的虚拟磁盘服务器的名称。
- **文件示例。**第二个示例是将文件导出为块设备。

```
primary$ ldm add-vdsdev backend voll@primary-vds0
```

其中：

- *backend* 是导出为块设备的实际文件的路径名。添加设备时，后端必须与设备名称成对出现。
- *voll* 是必须为添加到虚拟磁盘服务器的设备指定的唯一名称。该卷名称对于此虚拟磁盘服务器实例必须是唯一的，这是因为该名称将由此虚拟磁盘服务器导出到客户机以便进行添加。添加设备时，卷名称必须与实际设备的路径名成对出现。
- *primary-vds0* 是要将此设备添加到的虚拟磁盘服务器的名称。

6. 为来宾域添加虚拟磁盘。

以下示例会为来宾域 *ldg1* 添加虚拟磁盘。

```
primary$ ldm add-vdisk vdisk1 voll@primary-vds0 ldg1
```

其中：

- *vdisk1* 是虚拟磁盘的名称。
- *voll* 是要连接到的现有卷的名称。
- *primary-vds0* 是要连接到的现有虚拟磁盘服务器的名称。

注 – 虚拟磁盘是各种类型的物理设备、卷或文件所支持的通用块设备。虚拟磁盘并不与 SCSI 磁盘同义，因此磁盘标号中不包含目标 ID。逻辑域中的虚拟磁盘具有以下格式：cNdnNsN，其中 cN 是虚拟控制器，dN 是虚拟磁盘号，sN 是片。

7. 为来宾域设置 auto-boot 和 boot-device 变量。

第一个示例命令会将来宾域 ldg1 的 auto-boot\? 设置为 true。

```
primary$ ldm set-var auto-boot\?=true ldg1
```

第二个示例命令会将来宾域 ldg1 的 boot-device 设置为 vdisk。

```
primary$ ldm set-var boot-device=vdisk ldg1
```

8. 将资源绑定到来宾域 ldg1，然后列出该域以检验它是否已绑定。

```
primary$ ldm bind-domain ldg1
primary$ ldm list-domain ldg1
```

NAME	STATE	FLAGS	CONS	VCPU	MEMORY	UTIL	UPTIME
ldg1	bound	-----	5001	4	512M		

9. 要查找来宾域的控制台端口，可以查看上述 list-domain 子命令的输出。

可以看到在标题 Cons 下，逻辑域 guest 1 (ldg1) 的控制台输出已绑定到端口 5001。

10. 启动来宾域 ldg1。

```
primary$ ldm start-domain ldg1
```

11. 连接到来宾域的控制台。有多种方法可以完成此操作。

- 可以登录到控制域，然后直接连接到本地主机上的控制台端口：

```
$ ssh admin@controldom.domain
$ telnet localhost 5001
```

- 如果在 vntsd(1M) SMF 清单中启用了来宾控制台，也可以通过网络连接到来宾控制台。例如：

```
$ telnet host-name 5001
```

服务管理工具清单是一个描述服务的 XML 文件。有关创建 SMF 清单的更多信息，请参阅 Solaris 10 System Administrator Collection。

注 — 要通过控制台访问来宾域中的非英语 OS，控制台终端必须采用该 OS 所需的语言环境。

对来宾域进行 JumpStart 设置

如果您要对来宾域进行 JumpStart 设置，需要使用标准 JumpStart 过程，并进行下列配置文件语法更改，使其从常规的 Solaris OS JumpStart 过程变为特定于 LDom 的 JumpStart 过程，如以下两个示例中所示。

标准 JumpStart 配置文件

```
filesys c1t1d0s0 free /  
filesys c1t1d0s1 2048 swap  
filesys c1t1d0s5 120 /spare1  
filesys c1t1d0s6 120 /spare2
```

逻辑域中的虚拟磁盘设备名称与物理磁盘设备名称的不同之处在于，其设备名称中不包含目标 ID (tN)。虚拟磁盘设备名称并没有采用标准的 cNtNdNsN 格式，而是采用 cNdNsN 格式，其中 cN 是虚拟控制器，dN 是虚拟磁盘号，sN 是片。修改您的 JumpStart 配置文件以反映此更改，如以下配置文件示例中所示。

用于逻辑域的实际配置文件

```
filesys c0d0s0 free /  
filesys c0d0s1 2048 swap  
filesys c0d0s5 120 /spare1  
filesys c0d0s6 120 /spare2
```

第5章

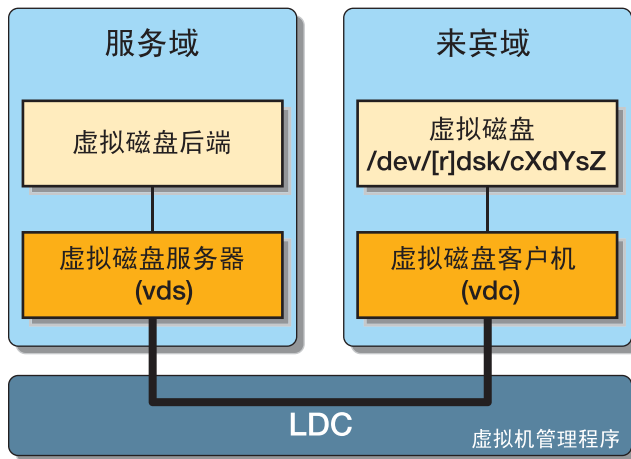
将虚拟磁盘与 Logical Domains 结合使用

本章介绍如何将虚拟磁盘与 Logical Domains 软件结合使用。

虚拟磁盘简介

一个虚拟磁盘包含两个组件：出现在来宾域中的虚拟磁盘本身，以及在其中存储数据和结束虚拟 I/O 的虚拟磁盘后端。虚拟磁盘后端由虚拟磁盘服务器 (virtual disk server, vds) 驱动程序从服务域中导出。vds 驱动程序通过虚拟机管理程序，借助于逻辑域通道 (logical domain channel, LDC) 与来宾域中的虚拟磁盘客户机 (virtual disk client, vdc) 驱动程序进行通信。最终，虚拟磁盘在来宾域中显示为 `/dev/[r]dsk/cXdYsZ` 设备。

图 5-1 虚拟磁盘与 Logical Domains



虚拟磁盘后端可以是物理磁盘、物理磁盘片、文件、卷管理框架（如 Zettabyte 文件系统 (Zettabyte File System, ZFS)、Solaris™ 卷管理器 (Solaris™ Volume Manager, SVM) 或 Veritas 卷管理器 (Veritas Volume Manager, VxVM)）中的卷或任何可从服务域访问的磁盘伪设备。

管理虚拟磁盘

本节介绍如何向来宾域中添加虚拟磁盘、如何更改虚拟磁盘和超时选项，以及如何从来宾域中删除虚拟磁盘。有关虚拟磁盘选项的说明，请参见第 53 页中的“虚拟磁盘后端选项”。有关虚拟磁盘超时的说明，请参见第 62 页中的“虚拟磁盘超时”。

▼ 添加虚拟磁盘

1. 使用以下命令从服务域中导出虚拟磁盘后端。

```
# ldm add-vdsdev [options={ro,slice,excl}] backend
volume_name@service_name
```

2. 使用以下命令将该后端指定给来宾域。

```
# ldm add-vdisk [timeout=seconds] disk_name volume_name@service_name ldom
```

注 — 在绑定来宾域 (ldom) 时，后端会实际从服务域中导出并指定给来宾域。

▼ 多次导出虚拟磁盘后端

可以通过相同或不同的虚拟磁盘服务器多次导出虚拟磁盘后端。随后，所导出的每个虚拟磁盘后端实例都可以指定给相同或不同的来宾域。

多次导出虚拟磁盘后端时不应使用独占 (`excl`) 选项。指定 `excl` 选项将只允许导出后端一次。使用 `ro` 选项，可以将后端作为只读设备安全地导出多次。



注意 – 多次导出虚拟磁盘后端时，在来宾域上运行且使用该虚拟磁盘的应用程序将负责协调和同步并发写入访问，以确保数据保持一致。

以下示例介绍如何通过同一个虚拟磁盘服务，将同一个虚拟磁盘添加到两个不同的来宾域中。

1. 使用以下命令从服务域中导出虚拟磁盘后端两次。

```
# ldm add-vdsdev [options={ro,slice}] backend volume1@service_name
# ldm add-vdsdev [options={ro,slice}] backend volume2@service_name
```

`add-vdsdev` 子命令显示以下警告，指示正在将后端导出多次。

```
Warning: "backend" is already in use by one or more servers in
guest "ldom"
```

2. 使用以下命令将所导出的后端指定给每个来宾域。

对于 `ldom1` 和 `ldom2` 来说，`disk_name` 可能会有所不同。

```
# ldm add-vdisk [timeout=seconds] disk_name volume1@service_name ldom1
# ldm add-vdisk [timeout=seconds] disk_name volume2@service_name ldom2
```

▼ 更改虚拟磁盘选项

- 从服务域中导出后端之后，可以使用以下命令来更改虚拟磁盘选项。

```
# ldm set-vdsdev options=[{ro,slice,excl}] volume_name@service_name
```

▼ 更改超时选项

- 将虚拟磁盘指定给来宾域之后，可以使用以下命令来更改虚拟磁盘的超时。

```
# ldm set-vdisk timeout=seconds disk_name ldom
```

▼ 删除虚拟磁盘

1. 使用以下命令从来宾域中删除虚拟磁盘。

```
# ldm rm-vdisk disk_name ldom
```

2. 使用以下命令停止从服务域中导出相应的后端。

```
# ldm rm-vdsdev volume_name@service_name
```

虚拟磁盘外观

将后端导出为虚拟磁盘之后，它可以在来宾域中显示为完整磁盘或具有单个片的磁盘。它的显示方式取决于后端的类型和用于导出后端的选项。

完整磁盘

将后端作为完整磁盘导出到某个域之后，它将在该域中显示为一个具有 8 个片（s0 至 s7）的常规磁盘。这样的磁盘可以通过 `format(1M)` 命令看到。可以使用 `fmthard(1M)` 或 `format(1M)` 命令更改该磁盘的分区表。

也可以在 OS 安装软件中看到完整磁盘，而且能够将其选作可从中安装 OS 的磁盘。

除了只能作为具有单个片的磁盘导出的物理磁盘片以外，所有的后端都可以作为完整磁盘导出。

具有单个片的磁盘

将后端作为具有单个片的磁盘导出到某个域之后，它将在该域中显示为一个具有单个分区 (s0) 的磁盘。这样的磁盘不能通过 `format(1M)` 命令看到，且不能更改其分区表。

不能从 OS 安装软件中看到具有单个片的磁盘，而且不能将其选作可从中安装 OS 的磁盘设备。

除了只能作为完整磁盘导出的物理磁盘以外，所有的后端都可以作为具有单个片的磁盘导出。

虚拟磁盘后端选项

导出虚拟磁盘后端时，可以指定不同的选项。这些选项在 `ldm add-vdsdev` 命令的 `options=` 参数中以逗号分隔的列表形式表示。有效选项包括：`ro`、`slice` 和 `excl`。

只读 (ro) 选项

只读 (`ro`) 选项指定将后端导出为只读设备。在这种情况下，只能对指定给来宾域的虚拟磁盘进行读取访问操作，对该虚拟磁盘进行的任何写入操作都将失败。

独占 (excl) 选项

独占 (`excl`) 选项指定在将服务域中的后端作为虚拟磁盘导出到另一个域之后，该后端必须由虚拟磁盘服务器以独占方式打开。以独占方式打开的后端将无法由服务域中的其他应用程序访问。这样可防止服务域中运行的应用程序无意中使用了正由来宾域使用的后端。

注 – 某些驱动程序不支持 `excl` 选项，而且将不允许以独占方式打开某些虚拟磁盘后端。已知 `excl` 选项适用于物理磁盘和片，但是该选项不适用于文件。该选项可能适用于也可能不适用于伪设备（如磁盘卷）。如果后端的驱动程序不支持以独占方式打开，则后端的 `excl` 选项将被忽略，而且后端将不以独占方式打开。

由于 `excl` 选项可防止服务域中运行的应用程序访问已导出到来宾域的后端，因此，在以下情况下，请勿设置 `excl` 选项：

- 当来宾域正在运行时，如果您希望能够使用某些命令（如 `format(1M)` 或 `luxadm(1M)`）来管理物理磁盘，则不要使用 `excl` 选项导出这些磁盘。
- 在导出 SVM 卷（如 RAID 卷或镜像卷）时，请不要设置 `excl` 选项。否则，在 RAID 卷或镜像卷的组件出现故障时，这可能会防止 SVM 启动相应的恢复操作。有关更多信息，请参见第 71 页中的“在 SVM 之上使用虚拟磁盘”。
- 如果 Veritas 卷管理器 (Veritas Volume Manager, VxVM) 安装在服务域中，而且针对物理磁盘启用了 Veritas 动态多路径 (Veritas Dynamic Multipathing, VxDMP)，则在导出物理磁盘时不得使用 `excl` 选项。否则，导出将失败，因为虚拟磁盘服务器 (virtual disk server, vds) 无法打开物理磁盘设备。有关更多信息，请参见第 71 页中的“在安装了 VxVM 的情况下使用虚拟磁盘”。
- 如果要从同一个虚拟磁盘服务多次导出同一个虚拟磁盘后端，请参见第 51 页中的“多次导出虚拟磁盘后端”了解更多信息。

默认情况下，后端以非独占方式打开。这样，在将后端导出到另一个域后，后端仍可以由服务域中运行的应用程序使用。请注意，这是从 Solaris 10 5/08 OS 发行版开始新增的行为。在 Solaris 10 5/08 OS 发行版之前，磁盘后端始终以独占方式打开，后端不可能以非独占方式打开。

片 (slice) 选项

通常，后端导出为完整磁盘或具有单个片的磁盘，具体取决于后端的类型。如果指定了 `slice` 选项，则会强制将后端导出为具有单个片的磁盘。

当您希望导出后端的原始内容时，此选项很有用。例如，如果您希望来宾域访问已经存储在 ZFS 或 SVM 卷中的数据，则应使用 `slice` 选项导出 ZFS 或 SVM 卷。

有关此选项的更多信息，请参见“虚拟磁盘后端”。

虚拟磁盘后端

虚拟磁盘后端是用来存储虚拟磁盘数据的位置。后端可以是磁盘、磁盘片、文件或卷（如 ZFS、SVM 或 VxVM）。后端在来宾域中显示为完整磁盘或具有单个片的磁盘，具体取决于从服务域中导出后端时是否设置了 `slice` 选项。默认情况下，虚拟磁盘后端以非独占方式导出为可读写的完整磁盘。

物理磁盘或磁盘 LUN

物理磁盘或逻辑单元号 (logical unit number, LUN) 始终导出为完整磁盘。在这种情况下，虚拟磁盘驱动程序 (vds 和 vdc) 从虚拟磁盘转发 I/O，并充当到物理磁盘或 LUN 的传递通道。

通过在不设置 slice 选项的情况下导出与磁盘的片 2 (s2) 相对应的设备，可以将物理磁盘或 LUN 从服务域中导出。如果使用 slice 选项导出磁盘的片 2，则将仅导出该片，而不是整个磁盘。

▼ 将物理磁盘导出为虚拟磁盘

1. 例如，要将物理磁盘 c1t48d0 导出为虚拟磁盘，则必须按如下方式从服务域中导出该磁盘的片 2 (c1t48d0s2)。

```
service# ldm add-vdsdev /dev/dsk/c1t48d0s2 c1t48d0@primary-vds0
```

2. 例如，从服务域中，将磁盘 (pdisk) 指定给来宾域 ldg1。

```
service# ldm add-vdisk pdisk c1t48d0@primary-vds0 ldg1
```

3. 来宾域启动且运行 Solaris OS 之后，可以列出磁盘（例如，c0d1），并查看该磁盘是否为可供访问的完整磁盘，即，具有 8 个片的常规磁盘。

```
ldg1# ls -l /dev/dsk/c0d1s*  
/dev/dsk/c0d1s0  
/dev/dsk/c0d1s1  
/dev/dsk/c0d1s2  
/dev/dsk/c0d1s3  
/dev/dsk/c0d1s4  
/dev/dsk/c0d1s5  
/dev/dsk/c0d1s6  
/dev/dsk/c0d1s7
```

物理磁盘片

物理磁盘片始终导出为具有单个片的磁盘。在这种情况下，虚拟磁盘驱动程序（vds 和 vdc）从虚拟磁盘转发 I/O，并充当到物理磁盘片的传递通道。

通过导出相应的片设备，可以将物理磁盘片从服务域中导出。如果该设备不同于片 2，那么，无论您是否指定了 slice 选项，该设备都会自动导出为具有单个片的磁盘。如果此设备是该磁盘的片 2，则必须设置 slice 选项才能仅将片 2 导出为具有单个片的磁盘；否则，整个磁盘将导出为一个完整磁盘。

▼ 将物理磁盘片导出为虚拟磁盘

1. 例如，要将物理磁盘 c1t57d0 的片 0 导出为虚拟磁盘，则必须按如下方式从服务域中导出与该片 (c1t57d0s0) 相对应的设备。

```
service# ldm add-vdsdev /dev/dsk/c1t57d0s0 c1t57d0s0@primary-vds0
```

您不必指定 slice 选项，因为片始终导出为具有单个片的磁盘。

2. 例如，从服务域中，将磁盘 (pslice) 指定给来宾域 ldg1。

```
service# ldm add-vdisk pslice c1t57d0s0@primary-vds0 ldg1
```

3. 来宾域启动且运行 Solaris OS 之后，可以列出磁盘（例如，c0d1），并查看该磁盘是否为可供访问且具有单个片的磁盘 (s0)。

```
ldg1# ls -l /dev/dsk/c0d13s*  
/dev/dsk/c0d13s0
```

▼ 导出片 2

- 要导出片 2（例如，磁盘 c1t57d0s2），必须指定 slice 选项；否则，将导出整个磁盘。

```
# ldm add-vdsdev options=slice /dev/dsk/c1t57d0s2 c1t57d0s2@primary-vds0
```

文件和卷

文件或卷（例如，来自 ZFS 或 SVM 中）导出为完整磁盘或具有单个片的磁盘，具体取决于是否设置了 `slice` 选项。

导出为完整磁盘的文件或卷

如果未设置 `slice` 选项，则文件或卷将导出为完整磁盘。在这种情况下，虚拟磁盘驱动程序（`vds` 和 `vdc`）从虚拟磁盘转发 I/O，并管理虚拟磁盘的分区。文件或卷最终会变成一个磁盘映像，其中包含虚拟磁盘所有片上的数据以及用于管理分区和磁盘结构的元数据。

将空白文件或卷导出为完整磁盘后，它在来宾域中显示为未格式化的磁盘（即，没有分区的磁盘）。然后，您需要在来宾域中运行 `format(1M)` 命令，以便定义可以使用的分区并编写有效的磁盘标签。在虚拟磁盘未进行格式化时，对该磁盘进行的任何 I/O 操作都将失败。

注 – 在 Solaris 10 5/08 OS 发行版之前，将空白文件导出为虚拟磁盘时，系统会编写一个默认磁盘标签并创建默认分区。从 Solaris 10 5/08 OS 发行版开始，将不再是这种情况，您必须在来宾域中运行 `format(1M)` 来创建分区。

▼ 将文件导出为完整磁盘

1. 从服务域中，创建一个要用作虚拟磁盘的文件（例如，`fdisk0`）。

```
service# mkfile 100m /ldoms/domain/test/fdisk0
```

文件的大小定义虚拟磁盘的大小。此示例创建一个 100 MB 的空白文件以获得一个 100 MB 的虚拟磁盘。

2. 从服务域中，将文件导出为虚拟磁盘。

```
service#ldm add-vdsdev /ldoms/domain/test/fdisk0
fdisk0@primary-vds0
```

在此示例中，未设置 `slice` 选项，因此该文件将导出为完整磁盘。

3. 例如，从服务域中，将磁盘 (`fdisk`) 指定给来宾域 `ldg1`。

```
service# ldm add-vdisk fdisk fdisk0@primary-vds0 ldg1
```

4. 来宾域启动且运行 Solaris OS 之后，可以列出磁盘（例如，c0d5），并查看该磁盘是否为可供访问的完整磁盘，即，具有 8 个片的常规磁盘。

```
ldg1# ls -l /dev/dsk/c0d5s*  
/dev/dsk/c0d5s0  
/dev/dsk/c0d5s1  
/dev/dsk/c0d5s2  
/dev/dsk/c0d5s3  
/dev/dsk/c0d5s4  
/dev/dsk/c0d5s5  
/dev/dsk/c0d5s6  
/dev/dsk/c0d5s7
```

导出为具有单个片的磁盘的文件或卷

如果设置了 slice 选项，则文件或卷将导出为具有单个片的磁盘。在这种情况下，虚拟磁盘仅具有一个分区 (s0)，该分区直接映射到文件或卷后端。该文件或卷仅包含写入虚拟磁盘的数据，而不包含额外的数据（如分区信息或磁盘结构）。

将文件或卷导出为具有单个片的磁盘时，系统会模拟一个假磁盘分区，这使得该文件或卷看上去像一个磁盘片。由于对磁盘分区进行了模拟，因此您不需要为该磁盘创建分区。

▼ 将 ZFS 卷导出为具有单个片的磁盘

1. 从服务域中，创建一个要用作具有单个片的磁盘的 ZFS 卷（例如，zdisk0）。

```
service# zfs create -V 100m ldoms/domain/test/zdisk0
```

卷的大小定义虚拟磁盘的大小。此示例创建一个 100 MB 的卷以获得一个 100 MB 的虚拟磁盘。

2. 从服务域中，导出与该 ZFS 卷相对应的设备，然后设置 slice 选项，以便将该卷导出为具有单个片的磁盘。

```
service# ldm add-vdsdev options=slice  
/dev/zvol/dsk/ldoms/domain/test/zdisk0 zdisk0@primary-vds0
```

3. 例如，从服务域中，将卷 (zdisk0) 指定给来宾域 ldg1。

```
service# ldm add-vdisk zdisk0 zdisk0@primary-vds0 ldg1
```


4. 来宾域启动且运行 **Solaris OS** 之后，可以列出磁盘（例如，c0d9），并查看该磁盘是否为可供访问且具有单个片的磁盘 (s0)。

```
ldg1# ls -l /dev/dsk/c0d9s*
/dev/dsk/c0d9s0
```

导出卷以及向后兼容性

在 Solaris 10 5/08 OS 发行版之前，slice 选项不存在，卷导出为具有单个片的磁盘。如果您有一个将卷导出为虚拟磁盘的配置，并且将系统升级到 Solaris 10 5/08 OS，那么，卷现在将导出为完整磁盘而不是具有单个片的磁盘。要保留旧行为并让卷导出为具有单个片的磁盘，需要执行下面的任一操作：

- 在 LDom 1.0.3 软件中使用 `ldm set-vdsdev` 命令，并针对要导出为具有单个片的磁盘的所有卷设置 `slice` 选项。有关此命令的更多信息，请参阅 `ldm` 手册页或《Logical Domains (LDoms) Manager 1.0.3 Man Page Guide》。
- 将下行添加到服务域上的 `/etc/system` 文件中。

```
set vds:vd_volume_force_slice = 1
```

注 – 设置这个可调参数会强制将所有卷导出为具有单个片的磁盘，并且您不能将任何卷导出为完整磁盘。

不同类型后端的导出方式汇总

后端	没有片选项	设置了片选项
磁盘（磁盘片 2）	完整磁盘*	具有单个片的磁盘 ^d
磁盘片（不是片 2）	具有单个片的磁盘\	具有单个片的磁盘
文件	完整磁盘	具有单个片的磁盘
卷（包括 ZFS、SVM 或 VxVM）	完整磁盘	具有单个片的磁盘

* 导出整个磁盘。
\ 片始终导出为具有单个片的磁盘。
^d 仅导出片 2

原则

使用回送文件 (lofi) 驱动程序

可以使用回送文件 (lofi) 驱动程序将文件导出为虚拟磁盘。但是，这样做会额外添加一个驱动程序层，并影响虚拟磁盘的性能。您可以改为直接将文件导出为完整磁盘或具有单个片的磁盘。请参见第 57 页中的“文件和卷”。

直接或间接导出磁盘片

要将片直接或间接（例如，通过 SVM 卷）导出为虚拟磁盘，请通过使用 `prtvtoc(1M)` 命令来确保该片不始于物理磁盘的第一个块（块 0）。

如果您直接或间接导出始于物理磁盘第一个块的磁盘片，则可能会覆写物理磁盘的分区表，并使该磁盘的所有分区不可访问。

CD、DVD 和 ISO 映像

可以按照与导出任何常规磁盘相同的方式来导出光盘 (compact disc, CD) 或数字通用光盘 (digital versatile disc, DVD)。要将 CD 或 DVD 导出到来宾域，请将 CD 或 DVD 设备的片 2 导出为完整磁盘，即，不使用 `slice` 选项。

注 – 不能导出 CD 或 DVD 驱动器本身，而只能导出位于 CD 或 DVD 驱动器内部的 CD 或 DVD。因此，CD 或 DVD 必须存在于驱动器内部，才能将其导出。同样，为了能够导出某个 CD 或 DVD，该 CD 或 DVD 不能在服务域中处于使用状态。特别是，卷管理文件系统 `volfs(7FS)` 服务不得使用该 CD 或 DVD。有关如何使设备不再被 `volfs` 使用的说明，请参见第 61 页中的“将 CD 或 DVD 从服务域导出到来宾域”。

如果具有存储在文件中或卷上的 CD 或 DVD 的国际标准化组织 (International Organization for Standardization, ISO) 映像，并将该文件或卷导出为完整磁盘，则该文件或卷将在来宾域中显示为一个 CD 或 DVD。

在导出 CD、DVD 或 ISO 映像时，它将自动在来宾域中显示为一个只读设备。但是，不能从来宾域中执行任何 CD 控制操作，即，不能从来宾域中启动、停止或弹出 CD。如果所导出的 CD、DVD 或 ISO 映像可引导，则可以在相应的虚拟磁盘上引导来宾域。

例如，如果导出 Solaris OS 安装 DVD，则可以在与该 DVD 相对应的虚拟磁盘上引导来宾域，并从该 DVD 安装来宾域。为此，当来宾域进入 `ok` 提示符时，请使用以下命令。

```
ok boot /virtual-devices@100/channel-devices@200/disk@n:f
```

其中，*n* 是表示所导出 DVD 的虚拟磁盘的索引。

注 – 如果您导出某个 Solaris OS 安装 DVD，并在与该 DVD 相对应的虚拟磁盘上引导一个来宾域以安装该来宾域，则在安装过程中不能更改该 DVD。因此，您可能需要跳过任何请求另一个 CD/DVD 的安装步骤，或者将需要提供一个替代路径来访问所请求的这个介质。

▼ 将 CD 或 DVD 从服务域导出到来宾域

1. 在 CD 或 DVD 驱动器中插入 CD 或 DVD。
2. 从服务域中，检查卷管理守护进程 **vold(1M)** 是否正在运行且处于联机状态。

```
service# svcs volfs
STATE          STIME      FMRI
online         12:28:12  svc:/system/filesystem/volfs:default
```

3. 执行以下操作之一。
 - 如果卷管理守护进程未在运行或未处于联机状态，请转至步骤 5。
 - 如果卷管理守护进程正在运行且处于联机状态（如步骤 2 中的示例所示），请执行以下操作：
 - a. 编辑 `/etc/vold.conf` 文件，并注释掉以下面的文字开头的行。

```
use cdrom drive....
```

有关更多信息，请参阅 **vold.conf(1M)** 手册页。

- b. 从服务域中，重新启动卷管理文件系统服务。

```
service# svcadm refresh volfs
service# svcadm restart volfs
```

4. 从服务域中，查找 CD-ROM 设备的磁盘路径。

```
service# cdwr -l
Looking for CD devices...
      Node                               Connected Device                               Device type
-----+-----+-----+-----+-----+-----+-----+-----+-----+
/dev/rdisk/c1t0d0s2 | MATSHITA CD-RW CW-8124 DZ13 | CD Reader/Writer
```

5. 从服务域中，将 CD 或 DVD 磁盘设备导出为一个完整磁盘。

```
service# ldm add-vdsdev /dev/dsk/c1t0d0s2 cdrom@primary-vds0
```

6. 从服务域中，将所导出的 CD 或 DVD 指定给来宾域（在此示例中为 ldg1）。

```
service# ldm add-vdisk cdrom cdrom@primary-vds0 ldg1
```

多次导出 CD 或 DVD

可以多次导出 CD 或 DVD 并将其指定给不同的来宾域。有关更多信息，请参见第 51 页中的“多次导出虚拟磁盘后端”。

虚拟磁盘超时

默认情况下，如果提供虚拟磁盘后端访问的服务域已关闭，从来宾域到相应虚拟磁盘的所有 I/O 都将被阻塞。当服务域正常运行而且正在为虚拟磁盘后端提供 I/O 请求服务时，该 I/O 会自动恢复。

不过，在某些情况下，文件系统或应用程序可能不希望 I/O 操作被阻塞，而是希望当服务域关闭时间过长时，该操作将失败并报告一个错误。现在，可以为每个虚拟磁盘设置一个连接超时期限，随后可将其用于在来宾域上的虚拟磁盘客户机和服务域上的虚拟磁盘服务器之间建立连接。达到该超时期限时，只要服务域关闭而且未在虚拟磁盘客户机和服务器之间重新建立连接，所有暂挂的 I/O 和所有新的 I/O 都将失败。

可以通过执行以下操作之一来设置此超时值：

- 使用 `ldm add-vdisk` 命令。

```
ldm add-vdisk timeout=seconds disk_name volume_name@service_name ldom
```

- 使用 `ldm set-vdisk` 命令。

```
ldm set-vdisk timeout=seconds disk_name ldom
```

以秒为单位指定超时值。如果超时值设置为 0，则超时设置处于禁用状态，而且 I/O 会在服务域关闭时被阻塞（这是默认设置和行为）。

或者，可以通过将下行添加到来宾域上的 `/etc/system` 文件中来设置超时值。

```
set vdc:vdc_timeout = seconds
```

注 – 如果设置了这个可调参数，则它会覆写使用 `ldm` CLI 完成的任何超时设置。同样，这个可调参数会为来宾域中的所有虚拟磁盘设置超时值。

虚拟磁盘和 SCSI

如果将物理 SCSI 磁盘或 LUN 导出为完整磁盘，则相应的虚拟磁盘支持用户 SCSI 命令界面 `uscsi(7D)` 和多主机磁盘控制操作 `mhd(7I)`。其他虚拟磁盘（如将文件或卷作为后端的虚拟磁盘）不支持这些界面。

因此，使用 SCSI 命令（如 `SVM metaset` 或 `Solaris Cluster shared devices`）的应用程序或产品功能只能与那些将物理 SCSI 磁盘作为后端的虚拟磁盘一起用在来宾域中。

注 – 对用作虚拟磁盘后端的物理 SCSI 磁盘或 LUN 进行管理的服务域可以有效地执行 SCSI 操作。特别是，该服务域设置了 SCSI 预留空间。因此，在服务域和来宾域中运行的应用程序不应向同一个物理 SCSI 磁盘发出 SCSI 命令；否则，这可能会导致意外的磁盘状态。

虚拟磁盘和 `format(1M)` 命令

在来宾域中，`format(1M)` 命令适用于作为完整磁盘导出的虚拟磁盘。具有单个片的磁盘不能通过 `format(1M)` 命令看到，而且不能更改此类磁盘的分区。

其后端为 SCSI 磁盘的虚拟磁盘支持所有的 `format(1M)` 子命令。其后端不是 SCSI 磁盘的虚拟磁盘不支持某些 `format(1M)` 子命令（如 `repair` 和 `defect`）。在这种情况下，`format(1M)` 的行为类似于集成开发环境 (integrated development environment, IDE) 磁盘的行为。

注 – 如果所选的虚拟磁盘具有可扩展固件接口 (extensible firmware interface, EFI) 磁盘标号，则 `format(1M)` 命令将失败。请参阅《Logical Domains (LDDoms) 1.0.3 发行说明》中的错误号 6363316。

将 ZFS 用于虚拟磁盘

本节中的下列主题介绍了将 Zettabyte 文件系统 (Zettabyte File System, ZFS) 用于逻辑域上的虚拟磁盘：

- [第 64 页中的“在 ZFS 卷之上创建虚拟磁盘”](#)
- [第 65 页中的“在虚拟磁盘上使用 ZFS”](#)
- [第 67 页中的“将 ZFS 用于引导磁盘”](#)

在 ZFS 卷之上创建虚拟磁盘

以下过程介绍了如何在服务域中创建 ZFS 卷，并允许其他域将该卷作为虚拟磁盘使用。在此示例中，服务域与控制域相同，名为 `primary`。来宾域名为 `ldg1`（举例来说）。每一步骤中的提示符显示了运行命令的域。

▼ 在 ZFS 卷之上创建虚拟磁盘

1. 创建 ZFS 存储池 (zpool)。

```
primary# zpool create -f tank1 c2t42d1
```

2. 创建 ZFS 卷。

```
primary# zfs create -V 100m tank1/myvol
```

3. 检验是否已创建 zpool（在此示例中为 `tank1`）和 ZFS 卷（在此示例中为 `tank1/myvol`）。

```
primary# zfs list
```

NAME	USED	AVAIL	REFER	MOUNTPOINT
tank1	100M	43.0G	24.5K	/tank1
tank1/myvol	22.5K	43.1G	22.5K	-

4. 配置服务，使 `tank1/myvol` 导出为虚拟磁盘。

```
primary# ldm add-vdsdev options=slice /dev/zvol/dsk/tank1/myvol  
zvol@primary-vds0
```

5. 将导出的磁盘添加到另一个域（在此示例中为 ldg1）。

```
primary# ldm add-vdisk vdisk zvol@primary-vds0 ldg1
```

6. 在另一个域（在此示例中为 ldg1）中，启动域并确保新的虚拟磁盘可见（可能需要运行 `devfsadm` 命令）。

在此示例中，新磁盘显示为 `/dev/rdisk/c2d2s0`。

```
ldg1# newfs /dev/rdisk/c2d2s0
newfs: construct a new file system /dev/rdisk/c2d2s0: (y/n)? y
Warning: 4096 sector(s) in last cylinder unallocated
Warning: 4096 sector(s) in last cylinder unallocated
/dev/rdisk/c2d2s0: 204800 sectors in 34 cylinders of 48 tracks, 128 sectors
100.0MB in 3 cyl groups (14 c/g, 42.00MB/g, 20160 i/g) super-block backups
(for fsck -F ufs -o b=#) at: 32, 86176, 172320,

ldg1# mount /dev/dsk/c2d2s0 /mnt

ldg1# df -h /mnt
Filesystem                size  used  avail capacity  Mounted on
/dev/dsk/c2d2s0            93M   1.0M   82M         2%   /mnt
```

注 – 在此示例中，ZFS 卷导出为具有单个片的磁盘。您也可以将 ZFS 卷导出为完整磁盘。如果将 ZFS 卷导出为完整磁盘，则可以对虚拟磁盘进行分区或者在虚拟磁盘上安装 Solaris OS。

在虚拟磁盘上使用 ZFS

以下过程说明了如何在域的虚拟磁盘之上直接使用 ZFS。可以使用 Solaris 10 OS `zpool(1M)` 和 `zfs(1M)` 命令，在虚拟磁盘之上创建 ZFS 池、文件系统和卷。尽管存储后端不同（虚拟磁盘，而不是物理磁盘），但 ZFS 的使用没有改变。

此外，如果已经具有 ZFS 文件系统，则可以将该文件系统从服务域导出，供在其他域中使用。

在此示例中，服务域与控制域相同，名为 `primary`。来宾域名为 `ldg1`（举例来说）。每一步骤中的提示符显示了运行命令的域。

▼ 在虚拟磁盘上使用 ZFS

1. 创建 ZFS 池（在此示例中为 tank），并检验该池是否已创建。

```
primary# zpool create -f tank c2t42d0
primary# zpool list
```

NAME	SIZE	USED	AVAIL	CAP	HEALTH	ALTROOT
tank	43.8G	108K	43.7G	0%	ONLINE	-

2. 创建 ZFS 文件系统（在此示例中为 tank/test），并检验该文件系统是否已创建。
在此示例中，通过在服务域上运行以下命令，在磁盘 c2t42d0 之上创建文件系统。

```
primary# zfs create tank/test
primary# zfs list
```

NAME	USED	AVAIL	REFER	MOUNTPOINT
tank	106K	43.1G	25.5K	/tank
tank/test	24.5K	43.1G	24.5K	/tank/test

3. 导出 ZFS 池（在此示例中为 tank）。

```
primary# zpool export tank
```

4. 配置服务，使物理磁盘 c2t42d0s2 导出为虚拟磁盘。

```
primary# ldm add-vdsdev /dev/rdisk/c2t42d0s2 volz@primary-vds0
```

5. 将导出的磁盘添加到另一个域（在此示例中为 ldg1）。

```
primary# ldm add-vdisk vdiskz volz@primary-vds0 ldg1
```


6. 在另一个域（在此示例中为 ldg1）中，启动域并确保新的虚拟磁盘可见（可能需要运行 devfsadm 命令），然后导入 ZFS 池。

```
ldg1# zpool import tank
ldg1# zpool list
NAME                SIZE      USED      AVAIL    CAP    HEALTH    ALTROOT
tank                43.8G    214K      43.7G    0%     ONLINE    -

ldg1# zfs list
NAME                USED      AVAIL    REFER    MOUNTPOINT
tank                106K      43.1G    25.5K    /tank
tank/test           24.5K      43.1G    24.5K    /tank/test

ldg1# df -hl -F zfs
Filesystem          size      used      avail    capacity    Mounted on
tank                43G       25K      43G       1%          /tank
tank/test           43G       24K      43G       1%          /tank/test
```

此时已导入 ZFS 池（在此示例中为 tank/test），并可在域 ldg1 中使用。

将 ZFS 用于引导磁盘

可以将带有大文件的 ZFS 文件系统用作逻辑域中的虚拟磁盘。

注 – ZFS 文件系统要求服务域中具有更多内存。在配置服务域时，应将这一点考虑进去。

ZFS 能够：

- 快速克隆文件系统
- 使用克隆置备更多的域
- 向文件上的磁盘以及 ZFS 文件系统文件进行网络安装

▼ 将 ZFS 用于引导磁盘

可以使用以下过程为逻辑域创建 ZFS 磁盘，也可以对这些磁盘实施快照和进行克隆，供其他域使用。

1. 在 primary 域中，保留整个磁盘或片以用作 ZFS 池的存储。步骤 2 将使用磁盘片 5。
2. 创建 ZFS 池；例如 ldomspool。

```
# zpool create ldomspool /dev/dsk/c0t0d0s5
```

3. 为第一个域（在此示例中为 ldg1）创建 ZFS 文件系统。

```
# zfs create ldomspool/ldg1
```

4. 创建要用作该域的磁盘的文件。

```
# mkfile 1G /ldomspool/ldg1/bootdisk
```

5. 将该文件指定为在创建域时要使用的设备。

```
primary# ldm add-vdsdev /ldomspool/ldg1/bootdisk voll@primary-vds0
primary# ldm add-vdisk vdisk1 voll@primary-vds0 ldg1
```

6. 引导域 ldg1，并对 vdisk1 进行网络安装。该文件可以用作完整的磁盘，并可以具有多个分区（即 root、usr、home、dump 和 swap 等各个分区）。
7. 完成安装后，对文件系统实施快照。

```
# zfs snapshot ldomspool/ldg1@initial
```

注 – 如果在重新引导域之前实施快照，则不会将域状态保存为快照或利用快照创建的任何其他克隆的一部分。

8. 利用快照创建更多克隆，并将其用作其他域（在此示例中为 ldg2 和 ldg3）的引导磁盘。

```
# zfs clone ldomspool/ldg1@initial ldomspool/ldg2
# zfs clone ldomspool/ldg1@initial ldomspool/ldg3
```

9. 检验每一项均已成功创建。

# zfs list					
NAME	USED	AVAIL	REFER	MOUNTPOINT	
ldomspool	1.07G	2.84G	28.5K	/ldomspool	
ldomspool/ldg1	1.03G	2.84G	1.00G	/ldomspool/ldg1	
ldomspool/ldg1@initial	23.0M	-	1.00G	-	
ldomspool/ldg2	23.2M	2.84G	1.00G	/ldomspool/ldg2	
ldomspool/ldg3	21.0M	2.84G	1.00G	/ldomspool/ldg3	

注 – 请确保 ZFS 池有足够的空间来容纳所创建的克隆。ZFS 使用写复制，仅当克隆中的块发生修改时，才使用 ZFS 池的空间。即使在引导域之后，克隆也仅使用磁盘的很少一部分空间（因为大部分 OS 二进制与初始快照中的 OS 二进制相同）。

在逻辑域环境中使用卷管理器

本节将介绍下列主题：

- [第 69 页](#)中的“在卷管理器之上使用虚拟磁盘”
- [第 72 页](#)中的“在虚拟磁盘之上使用卷管理器”

在卷管理器之上使用虚拟磁盘

可以将任何 Zettabyte 文件系统 (Zettabyte File System, ZFS)、Solaris 卷管理器 (Solaris Volume Manager, SVM) 或 Veritas 卷管理器 (Veritas Volume Manager, VxVM) 卷作为虚拟磁盘从服务域导出到来宾域。卷可以导出为具有单个片的磁盘（如果在 `ldm add-vdsdev` 命令中指定了 `slice` 选项）或完整磁盘。

注 – 本节的其余部分将以 SVM 卷为例。但是，所讨论的内容也适用于 ZFS 卷和 VxVM 卷。

以下示例说明如何将卷导出为具有单个片的磁盘。例如，如果服务域将 SVM 卷 `/dev/md/dsk/d0` 作为具有单个片的磁盘导出到 `domain1`，而且 `domain1` 将该虚拟磁盘视为 `/dev/dsk/c0d2*`，则 `domain1` 中只有一个 `s0` 设备，即 `/dev/dsk/c0d2s0`。

来宾域中的虚拟磁盘（例如 `/dev/dsk/c0d2s0`）直接映射到相关联的卷（例如 `/dev/md/dsk/d0`），来宾域的虚拟磁盘上存储的数据直接存储到相关联的卷中，而没有额外的元数据。因此，也可以通过相关联的卷从服务域直接访问来宾域的虚拟磁盘上存储的数据。

示例：

- 如果将 SVM 卷 d0 从 primary 域导出到 domain1，则 domain1 配置需要一些额外的步骤。

```
primary# metainit d0 3 1 c2t70d0s6 1 c2t80d0s6 1 c2t90d0s6
primary# ldm add-vdsdev options=slice /dev/md/dsk/d0
vol3@primary-vds0
primary# ldm add-vdisk vdisk3 vol3@primary-vds0 domain1
```

- 绑定并启动 domain1 后，导出的卷显示为 `/dev/dsk/c0d2s0`（举例来说），并且您可以使用它。

```
domain1# newfs /dev/rdisk/c0d2s0
domain1# mount /dev/dsk/c0d2s0 /mnt
domain1# echo test-domain1 > /mnt/file
```

- 停止并取消绑定 domain1 后，可以通过 SVM 卷 d0 从主域直接访问 domain1 的虚拟磁盘上存储的数据。

```
primary# mount /dev/md/dsk/d0 /mnt
primary# cat /mnt/file
test-domain1
```

注 – 具有单个片的磁盘不能通过 `format(1M)` 命令看到，不能进行分区，也不能用作 Solaris OS 的安装磁盘。有关此主题的更多信息，请参见第 52 页中的“[虚拟磁盘外观](#)”。

在 SVM 之上使用虚拟磁盘

当 RAID 或镜像 SVM 卷由另一个域用作虚拟磁盘时，则在导出它时不得设置独占 (excl) 选项。否则，如果 SVM 卷的某个组件出现故障，则无法启动使用 `metareplace` 命令或使用热备件恢复 SVM 卷的过程。`metastat` 命令将卷视为正在进行重新同步，但并没有进行重新同步。

例如，`/dev/md/dsk/d0` 是通过 `excl` 选项作为虚拟磁盘导出到另一个域的 RAID SVM 卷，并且 `d0` 配有一些热备用设备。如果 `d0` 的组件出现故障，则 SVM 会使用热备件来替换故障组件，并对 SVM 卷进行重新同步。但是，重新同步并不会开始。卷会被报告为正在进行重新同步，但并没有进行重新同步。

```
# metastat d0
d0: RAID
  State: Resyncing
  Hot spare pool: hsp000
  Interlace: 32 blocks
  Size: 20097600 blocks (9.6 GB)
Original device:
  Size: 20100992 blocks (9.6 GB)
Device                               Start Block  Dbase   State Reloc
c2t2d0s1                             330         No      Okay   Yes
c4t12d0s1                             330         No      Okay   Yes
/dev/dsk/c10t600C0FF0000000000015153295A4B100d0s1 330         No      Resyncing Yes
```

这种情况下，必须停止并取消绑定将 SVM 卷用作虚拟磁盘的域，以便完成重新同步。随后可以使用 `metasync` 命令重新同步 SVM 卷。

```
# metasync d0
```

在安装了 VxVM 的情况下使用虚拟磁盘

如果在系统上安装了 Veritas 卷管理器 (Veritas Volume Manager, VxVM)，而且针对要导出为虚拟磁盘的物理磁盘或分区启用了 Veritas 动态多路径 (Dynamic Multipathing, DMP)，则必须在未设置 `excl` 选项的情况下导出该磁盘或分区。否则，在绑定使用此类磁盘的域时，您会收到 `/var/adm/messages` 中的错误。

```
vd_setup_vd(): ldi_open_by_name(/dev/dsk/c4t12d0s2) = errno 16
vds_add_vd(): Failed to add vdisk ID 0
```

通过查看 `vxdisk list` 命令输出中的多路径信息，可确认是否启用了 Veritas DMP；例如：

```
# vxdisk list Disk_3
Device:      Disk_3
devicetag:   Disk_3
type:        auto
info:         format=none
flags:        online ready private autoconfig invalid
pubpaths:    block=/dev/vx/dmp/Disk_3s2 char=/dev/vx/rdmp/Disk_3s2
guid:         -
udid:         SEAGATE%5FST336753LSUN36G%5FDISKS%5F3032333948303144304E0000
site:         -
Multipathing information:
numpaths:     1
c4t12d0s2    state=enabled
```

或者，如果您针对要在设置了 `excl` 选项的情况下导出为虚拟磁盘的磁盘或片启用了 Veritas DMP，则可以使用 `vxddmpadm` 命令禁用 DMP。例如：

```
# vxddmpadm -f disable path=/dev/dsk/c4t12d0s2
```

在虚拟磁盘之上使用卷管理器

本节介绍了 Logical Domains 环境中的下列情形：

- [第 72 页中的“在虚拟磁盘之上使用 ZFS”](#)
- [第 73 页中的“在虚拟磁盘之上使用 SVM”](#)
- [第 73 页中的“在虚拟磁盘之上使用 VxVM”](#)

在虚拟磁盘之上使用 ZFS

任何虚拟磁盘都可以与 ZFS 配合使用。可以在任何域中导入 ZFS 存储池 (zpool)，该域可看到属于 zpool 的所有存储设备，无论该域是将所有这些设备视为虚拟设备还是实际设备。

在虚拟磁盘之上使用 SVM

在 SVM 本地磁盘集中可以使用任何虚拟磁盘。例如，可以使用虚拟磁盘来存储本地磁盘集的 SVM 元设备状态数据库 `metadb(1M)`，或在本地磁盘集中创建 SVM 卷。

其后端为 SCSI 磁盘的任何虚拟磁盘都可以用在 SVM 共享磁盘集 `metaset(1M)` 中。其后端不是 SCSI 磁盘的虚拟磁盘不能添加到 SVM 共享磁盘集中。如果尝试将其后端不是 SCSI 磁盘的虚拟磁盘添加到 SVM 共享磁盘集，则会失败并显示类似以下内容的错误。

```
# metaset -s test -a c2d2
metaset: domain1: test: failed to reserve any drives
```

在虚拟磁盘之上使用 VxVM

有关来宾域的 VxVM 支持信息，请参阅 Symantec 的 VxVM 文档。

第6章

其他信息和任务

本章包含有关使用前面章节中未介绍的 Logical Domains 软件的信息和任务。

在 CLI 中输入名称的限制

以下各节介绍了在 Logical Domains Manager CLI 中输入名称的限制。

文件名称 (*file*) 和变量名称 (*var_name*)

- 第一个字符必须是字母、数字或斜杠 (/)。
- 后面的字符必须是字母、数字或标点。

虚拟磁盘服务器 *backend* 和虚拟交换机设备名称

- 必须包含字母、数字或标点。

配置名称 (*config_name*)

您为存储在系统控制器上的配置指定的逻辑域配置名称 (*config_name*) 不得超过 64 个字符。

所有其他名称

其余名称，例如逻辑域名称 (*ldom*)、服务名称 (*vswitch_name*、*service_name*、*vdpcs_service_name* 和 *vcc_name*)、虚拟网络名称 (*if_name*) 以及虚拟磁盘名称 (*disk_name*)，必须采用以下格式：

- 第一个字符必须是字母或数字。
- 后面的字符必须是字母、数字或以下任意字符：‘_-+#.~;~()’

使用 `ldm list` 子命令

本节显示了 `ldm` 子命令的语法用法，定义了一些输出项（例如，标志和使用率统计信息），并提供了输出示例。

计算机可读的输出

如果您要创建使用 `ldm list` 命令输出的脚本，请始终使用 `-p` 选项生成计算机可读形式的输出。有关更多信息，请参见第 84 页中的“生成可解析的、计算机可读的列表 (`-p`)”。

▼ 显示 `ldm` 子命令的语法用法

- 要查看所有 `ldm` 子命令的语法用法，请执行以下操作。

代码示例 6-1 所有 `ldm` 子命令的语法用法

```
primary# ldm --help

Usage:
  ldm [--help] command [options] [properties] operands

Command(s) for each resource (aliases in parens):

  bindings
    list-bindings [-e] [-p] [<ldom>...]

  services
    list-bindings [-e] [-p] [<ldom>...]

  constraints
    list-constraints ([-x] | [-e] [-p]) [<ldom>...]
```

```

devices
    list-devices [-a] [-p] [cpu] [crypto|mau] [memory] [io]

domain      ( dom )
    add-domain (-i <file> | mac-addr=<num> <ldom> | <ldom>...)
    remove-domain (-a | <ldom>...)
    list-domain [-e] [-l] [-p] [<ldom>...]
    start-domain start-domain (-a | -i <file> | <ldom>...)
    stop-domain stop-domain [-f] (-a | <ldom>...)
    bind-domain (-i <file> | <ldom>)
    unbind-domain <ldom>
    panic-domain <ldom>

io
    add-io [bypass=on] <bus> <ldom>
    remove-io <bus> <ldom>

crypto      ( mau )
    add-crypto <number> <ldom>
    set-crypto <number> <ldom>
    remove-crypto <number> <ldom>

memory      ( mem )
    add-memory <number>[GMK] <ldom>
    set-memory <number>[GMK] <ldom>
    remove-memory <number>[GMK] <ldom>

reconf
    remove-reconf <ldom>

spconfig    ( config )
    add-spconfig <config_name>
    set-spconfig <config_name>
    remove-spconfig <config_name>
    list-spconfig

variable    ( var )
    add-variable <var_name>=<value>... <ldom>
    set-variable <var_name>=<value>... <ldom>
    remove-variable <var_name>... <ldom>
    list-variable [<var_name>...] <ldom>

vconscon    ( vcc )
    add-vconscon port-range=<x>-<y> <vcc_name> <ldom>
    set-vconscon port-range=<x>-<y> <vcc_name>
    remove-vconscon [-f] <vcc_name>

```

```

vconsole      ( vcons )
    set-vcons [port=[<port-num>]] [group=<group>] [service=<vcc_server>]
<ldom>

vcpu
    add-vcpu <number> <ldom>
    set-vcpu <number> <ldom>
    remove-vcpu <number> <ldom>

vdisk
    add-vdisk [timeout=<seconds>] <disk_name>
<volume_name>@<service_name> <ldom>
    set-vdisk [timeout=<seconds>] [volume=<volume_name>@<service_name>]
<disk_name> <ldom>
    remove-vdisk [-f] <disk_name> <ldom>

vdiskserver ( vds )
    add-vdiskserver <service_name> <ldom>
    remove-vdiskserver [-f] <service_name>

vdpcc         ( ndpsldcc )
    add-vdpcc <vdppcc_name> <service_name> <ldom>
    remove-vdpcc [-f] <vdppcc_name> <ldom>

vdpcs         ( ndpsldcs )
    add-vdpcs <vdppcs_name> <ldom>
    remove-vdpcs [-f] <vdppcs_name>

vdiskserverdevice ( vdsdev )
    add-vdiskserverdevice [options={ro,slice,excl}] <backend>
<volume_name>@<service_name>
    set-vdiskserverdevice options=[{ro,slice,excl}]
<volume_name>@<service_name>
    remove-vdiskserverdevice [-f] <volume_name>@<service_name>

vnet
    add-vnet [mac-addr=<num>] <if_name> <vswitch_name> <ldom>
    set-vnet [mac-addr=<num>] [vswitch=<vswitch_name>] <if_name> <ldom>
    remove-vnet [-f] <if_name> <ldom>

vswitch       ( vsw )
    add-vswitch [mac-addr=<num>] [net-dev=<device>] [mode=<mode>]
<vswitch_name> <ldom>
    set-vswitch [mac-addr=<num>] [net-dev=<device>] [mode=<mode>]
<vswitch_name>
    remove-vswitch [-f] <vswitch_name>

```

代码示例 6-1 所有 ldm 子命令的语法用法（续）

Verb aliases:		
	Alias	Verb
	-----	-----
	rm	remove
	ls	list
Command aliases:		
	Alias	Command
	-----	-----
	create	add-domain
	destroy	remove-domain
	cancel-reconf	remove-reconf
	start	start-domain
	stop	stop-domain
	bind	bind-domain
	unbind	unbind-domain
	panic	panic-domain

标志定义

在域的输出中可以显示以下标志。

-	占位符
c	控制域
d	延迟重新配置
n	正常
s	正在启动或停止
t	转换
v	虚拟 I/O 域

如果使用命令的长 (-l) 选项，则拼写出标志。否则，将显示字母缩写。

列表标志值与位置相关。在从左至右排列的五列中的每一列中可以出现以下值。

第 1 列	第 2 列	第 3 列	第 4 列	第 5 列
s 或 -	n 或 t	d 或 -	c 或 -	v 或 -

使用率统计信息定义

在 `ldm list` 命令中使用长 (`-l`) 选项, 可以显示每个虚拟 CPU 的使用率统计信息 (UTIL)。统计信息是自上次显示统计信息以后的时间百分比, 即, 虚拟 CPU 代表来宾操作系统执行所用的时间。除非虚拟 CPU 被移交给虚拟机管理程序, 否则将其视为代表来宾操作系统执行。如果来宾操作系统没有将虚拟 CPU 移交给虚拟机管理程序, 则来宾操作系统中的 CPU 使用率始终显示为 100%。

为逻辑域报告的使用率统计信息是域中虚拟 CPU 的虚拟 CPU 使用率平均值。

各种列表示例

▼ 显示软件版本 (`-v`)

- 要查看当前安装的软件版本, 请执行以下操作, 所显示的列表与以下内容类似。

代码示例 6-2 安装的软件版本

```
primary$ ldm -v

Logical Domain Manager (v 1.0.3)
  Hypervisor control protocol v 1.0

System PROM:
  Hypervisor    v. 1.5.2          @(#)Hypervisor 1.5.2 2007/09/25 08:39/015
  OpenBoot      v. 4.27.2        @(#)OBP 4.27.2 2007/09/24 16:28
```

▼ 生成短列表

- 要生成所有域的短列表, 请执行以下操作。

代码示例 6-3 所有域的短列表

```
primary$ ldm list
```

NAME	STATE	FLAGS	CONS	VCPU	MEMORY	UTIL	UPTIME
primary	active	-t-cv		4	1G	0.5%	3d 21h 7m
ldg1	active	-t---	5000	8	1G	23%	2m

▼ 生成长列表 (-1)

- 要生成所有域的长列表，请执行以下操作。

代码示例 6-4 所有域的长列表

```
primary$ ldm list -l
NAME                STATE    FLAGS    CONS    VCPU    MEMORY    UTIL    UPTIME
primary             active   -t-cv                    1      768M      0.0%    0s

VCPU
  VID      PID      UTIL  STRAND
  0         0         0.0%   100%

MEMORY
  RA              PA              SIZE
  0x4000000       0x4000000       768M

IO
  DEVICE          PSEUDONYM      OPTIONS
  pci@780         bus_a
  pci@7c0         bus_b          bypass=on

VCC
  NAME            PORT-RANGE
  vcc0            5000-5100

VSW
  NAME            MAC              NET-DEV    DEVICE    MODE
  vsw0            08:00:20:aa:bb:e0 e1000g0    switch@0   prog,promisc
  vsw1            08:00:20:aa:bb:e1                    routed

VDS
  NAME            VOLUME          OPTIONS      DEVICE
  vds0            myvol-a         slice        /disk/a
                  myvol-b                             /disk/b
                  myvol-c         ro,slice,excl /disk/c
  vds1            myvol-d                             /disk/d

VDPCS
  NAME
  vdpes0
  vdpes1

-----
NAME                STATE    FLAGS    CONS    VCPU    MEMORY    UTIL    UPTIME
ldg1                bound    -----    5000     1      512M

VCPU
```

代码示例 6-4 所有域的长列表（续）

VID	PID	UTIL	STRAND
0	1	100%	
MEMORY			
RA	PA	SIZE	
0x4000000	0x34000000	512M	
NETWORK			
NAME	SERVICE	DEVICE	MAC
mynet-b	vsw0@primary	network@0	08:00:20:ab:9a:12
mynet-a	vsw0@primary	network@1	08:00:20:ab:9a:11
DISK			
NAME	VOLUME	DEVICE	SERVER
mydisk-a	myvol-a@vds0	disk@0	primary
mydisk-b	myvol-b@vds0	disk@1	primary
VDPCC			
NAME	SERVICE		
myvdpcc-a	vdpcs0@primary		
myvdpcc-b	vdpcs0@primary		
VCONS			
NAME	SERVICE	PORT	
mygroup	vcc0@primary	5000	

▼ 生成扩展列表 (-e)

- 要生成所有域的扩展列表，请执行以下操作。

代码示例 6-5 所有域的扩展列表

primary\$ ldm list -e							
NAME	STATE	FLAGS	CONS	VCPU	MEMORY	UTIL	UPTIME
primary	active	-t-cv		1	768M	0.0%	0s
VCPU							
VID	PID	UTIL	STRAND				
0	0	0.0%	100%				
MEMORY							
RA	PA	SIZE					
0x4000000	0x4000000	768M					
IO							
DEVICE	PSEUDONYM	OPTIONS					

代码示例 6-5 所有域的扩展列表（续）

pci@780		bus_a					
pci@7c0		bus_b		bypass=on			
VLDC							
NAME							
primary							
VCC							
NAME		PORT-RANGE					
vcc0		5000-5100					
VSW							
NAME		MAC	NET-DEV		DEVICE	MODE	
vsw0		08:00:20:aa:bb:e0	e1000g0		switch@0	prog,promisc	
vsw1		08:00:20:aa:bb:e1				routed	
VDS							
NAME		VOLUME	OPTIONS		DEVICE		
vds0		myvol-a	slice		/disk/a		
		myvol-b			/disk/b		
		myvol-c	ro,slice,excl		/disk/c		
vds1		myvol-d			/disk/d		
VDPCS							
NAME							
vdpcs0							
vdpcs1							
VLDCC							
NAME		SERVICE			DESC		
hvctl		primary@primary			hvctl		
vldcc0		primary@primary			ds		

NAME		STATE	FLAGS	CONS	VCPU	MEMORY	UTIL UPTIME
ldg1		bound	-----	5000	1	512M	
VCPU							
VID	PID	UTIL	STRAND				
0	1	100%					
MEMORY							
RA		PA		SIZE			
0x4000000		0x34000000		512M			
VLDCC							
NAME		SERVICE			DESC		

代码示例 6-5 所有域的扩展列表（续）

vldcc0	primary@primary	ds	
NETWORK			
NAME	SERVICE	DEVICE	MAC
myinet-b	vsw0@primary	network@0	08:00:20:ab:9a:12
myinet-a	vsw0@primary	network@1	08:00:20:ab:9a:11
DISK			
NAME	VOLUME	DEVICE	SERVER
mydisk-a	myvol-a@vds0	disk@0	primary
mydisk-b	myvol-b@vds0	disk@1	primary
VDPCC			
NAME	SERVICE		
myvdpcc-a	vdpcs0@primary		
myvdpcc-b	vdpcs0@primary		
VCONS			
NAME	SERVICE	PORT	
mygroup	vcc0@primary	5000	

▼ 生成可解析的、计算机可读的列表 (-p)

- 要生成所有域的可解析的、计算机可读的列表，请执行以下操作。

代码示例 6-6 计算机可读的列表

```
primary$ ldm list -p
VERSION 1.0
DOMAIN|name=primary|state=active|flags=-t-cv|cons=|ncpu=1|mem=805306368|util=
0.0|uptime=0
DOMAIN|name=ldg1|state=bound|flags=-----|cons=5000|ncpu=1|mem=536870912|util=
|uptime=
```

▼ 显示域的状态

- 要查看域（例如，来宾域 ldg1）的状态，请执行以下操作。

代码示例 6-7 域状态

```
primary# ldm list-domain ldg1
NAME          STATE  FLAGS  CONS  VCPU  MEMORY  UTIL  UPTIME
ldg1          active -t---  5000   8     1G     0.3%  2m
```

▼ 列出变量

- 要列出某个域（例如 ldg1）的变量（例如 boot-device），请执行以下操作。

代码示例 6-8 域的变量列表

```
primary$ ldm list-variable boot-device ldg1
boot-device=/virtual-devices@100/channel-devices@200/disk@0:a
```

▼ 列出绑定

- 要列出绑定到某个域（例如 ldg1）的资源，请执行以下操作。

代码示例 6-9 域的绑定列表

```
primary$ ldm list-bindings ldg1
NAME                STATE    FLAGS    CONS    VCPU    MEMORY    UTIL    UPTIME
ldg1                 bound    -----    5000     1       512M

VCPU
  VID    PID    UTIL  STRAND
  0       1      100%

MEMORY
  RA                PA                SIZE
  0x4000000         0x34000000       512M

NETWORK
  NAME                SERVICE                DEVICE    MAC
  mynet-b             vsw0@primary           network@0 08:00:20:ab:9a:12
    PEER
    vsw0@primary      08:00:20:aa:bb:e0
  mynet-a@ldg1        08:00:20:ab:9a:11
  mynet-c@ldg2        08:00:20:ab:9a:22
  NAME                SERVICE                DEVICE    MAC
  mynet-a             vsw0@primary           network@1 08:00:20:ab:9a:11
    PEER
    vsw0@primary      08:00:20:aa:bb:e0
  mynet-b@ldg1        08:00:20:ab:9a:12
  mynet-c@ldg2        08:00:20:ab:9a:22

DISK
  NAME                VOLUME                DEVICE    SERVER
  mydisk-a            myvol-a@vds0          disk@0    primary
  mydisk-b            myvol-b@vds0          disk@1    primary

VDPCC
  NAME                SERVICE
```

代码示例 6-9 域的绑定列表（续）

myvdpcc-a	vdpcs0@primary	
myvdpcc-b	vdpcs0@primary	
VCONS		
NAME	SERVICE	PORT
mygroup	vcc0@primary	5000

▼ 列出配置

- 要列出存储在 SC 上的逻辑域配置，请执行以下操作。

代码示例 6-10 配置列表

```
primary$ ldm list-config
factory-default [current]
initial [next]
```

标签的含义

配置名称右侧的标签具有以下含义：

- current - 当前正在使用的配置
- next - 在下一个关机循环时要使用的配置

▼ 列出设备

- 要列出所有服务器资源（绑定资源和非绑定资源），请执行以下操作。

代码示例 6-11 所有服务器资源列表

```
primary$ ldm list-devices -a
VCPU
  PID  %FREE
  0      0
  1      0
  2      0
  3      0
  4     100
  5     100
  6     100
  7     100
  8     100
  9     100
 10     100
 11     100
```

代码示例 6-11 所有服务器资源列表（续）

12	100		
13	100		
14	100		
15	100		
16	100		
17	100		
18	100		
19	100		
20	100		
21	100		
22	100		
23	100		
24	100		
25	100		
26	100		
27	100		
28	100		
29	100		
30	100		
31	100		
MAU			
CPUSET		BOUND	
(0, 1, 2, 3)		ldg2	
(4, 5, 6, 7)			
(8, 9, 10, 11)			
(12, 13, 14, 15)			
(16, 17, 18, 19)			
(20, 21, 22, 23)			
(24, 25, 26, 27)			
(28, 29, 30, 31)			
MEMORY			
PA	SIZE	BOUND	
0x0	512K	_sys_	
0x80000	1536K	_sys_	
0x200000	62M	_sys_	
0x4000000	768M	primary	
0x34000000	512M	ldg1	
0x54000000	8M	_sys_	
0x54800000	2G	ldg2	
0xd4800000	29368M		
IO			
DEVICE	PSEUDONYM	BOUND	OPTIONS
pci@780	bus_a	yes	
pci@7c0	bus_b	yes	bypass=on

▼ 列出服务

- 要列出可用的服务，请执行以下操作。

代码示例 6-12 服务列表

primary\$ ldm list-services				
VDS				
	NAME	VOLUME	OPTIONS	DEVICE
	primary-vds0			
VCC				
	NAME	PORT-RANGE		
	primary-vcc0	5000-5100		
VSW				
	NAME	MAC	NET-DEV	DEVICE MODE
	primary-vsw0	00:14:4f:f9:68:d0	e1000g0	switch@0 prog,promisc

列出约束

对于 Logical Domains Manager，约束是您希望分配给特定域的一个或多个资源。您会接收到要求添加到域中的所有资源，或者任何资源也得不到，这取决于可用资源。
list-constraints 子命令可列出您要求分配给域的那些资源。

▼ 列出一个域的约束

- 要列出一个域（例如 ldg1）的约束，请执行以下操作。

代码示例 6-13 一个域的约束列表

```
primary$ ldm list-constraints ldg1
DOMAIN
ldg1

VCPU
    COUNT
    1

MEMORY
    SIZE
    512M

NETWORK
    NAME          SERVICE          DEVICE          MAC
mynet-b          vsw0              network@0        08:00:20:ab:9a:12
mynet-b          vsw0              network@0        08:00:20:ab:9a:12
```

代码示例 6-13 一个域的约束列表（续）

DISK	
NAME	VOLUME
mydisk-a	myvol-a@vds0
mydisk-b	myvol-b@vds0
VDPCC	
NAME	SERVICE
myvdpcc-a	vdpcs0@primary
myvdpcc-b	vdpcs0@primary
VCONS	
NAME	SERVICE
mygroup	vcc0

▼ 以 XML 格式列出约束

- 要以 XML 格式列出特定域（例如 ldg1）的约束，请执行以下操作。

代码示例 6-14 某个域的约束（采用 XML 格式）

```
primary$ ldm list-constraints -x ldg1
<?xml version="1.0"?>
<LDM_interface version="1.0">
  <data version="2.0">
    <ldom>
      <ldom_info>
        <ldom_name>ldg1</ldom_name>
      </ldom_info>
      <cpu>
        <number>8</number>
      </cpu>
      <memory>
        <size>1G</size>
      </memory>
      <network>
        <vnet_name>vnet0</vnet_name>
        <service_name>primary-vsw0</service_name>
        <mac_address>01:14:4f:fa:0f:55</mac_address>
      </network>
      <disk>
        <vdisk_name>vdisk0</vdisk_name>
        <service_name>primary-vds0</service_name>
        <vol_name>vol0</vol_name>
      </disk>
      <var>
        <name>boot-device</name>
```

代码示例 6-14 某个域的约束（采用 XML 格式）（续）

```
<value>/virtual-devices@100/channel-devices@200/disk@0:a</value>
</var>
<var>
  <name>nvrarc</name>
  <value>devalias vnet0 /virtual-devices@100/channel-devices@200/
network@0</value>
</var>
<var>
  <name>use-nvrarc?</name>
  <value>true</value>
</var>
</ldom>
</data>
</LDM_interface>
```

▼ 以计算机可读格式列出约束

- 要以可解析的格式列出所有域的约束，请执行以下操作。

代码示例 6-15 所有域的约束（采用计算机可读的格式）

```
primary$ ldm list-constraints -p
VERSION 1.0
DOMAIN|name=primary
MAC|mac-addr=00:03:ba:d8:b1:46
VCPU|count=4
MEMORY|size=805306368
IO
|dev=pci@780|alias=
|dev=pci@7c0|alias=
VDS|name=primary-vds0
|vol=disk-ldg2|opts=|dev=/ldoms/nv72-ldg2/disk
|vol=vol0|opts=|dev=/ldoms/nv72-ldg1/disk
VCC|name=primary-vcc0|port-range=5000-5100
VSW|name=primary-vsw0|mac-addr=|net-dev=e1000g0|dev=switch@0
DOMAIN|name=ldg1
VCPU|count=8
MEMORY|size=1073741824
VARIABLES
|boot-device=/virtual-devices@100/channel-devices@200/disk@0:a
|nvrarc=devalias vnet0 /virtual-devices@100/channel-devices@200/network@0
|use-nvrarc?=true
VNET|name=vnet0|dev=network@0|service=primary-vsw0|mac-addr=01:14:4f:fa:0f:55
VDISK|name=vdisk0|vol=vol0@primary-vds0
```


ldm stop-domain 命令在域负载较重时会超时

ldm stop-domain 命令会在域完成关闭操作之前超时。如果发生这种情况，Logical Domains Manager 会返回一个类似以下内容的错误。

```
LDom ldg8 stop notification failed
```

但是，域可能仍在处理关闭请求。使用 ldm list-domain 命令可检验域的状态。例如：

```
# ldm list-domain ldg8
NAME          STATE  FLAGS  CONS  VCPU  MEMORY  UTIL  UPTIME
ldg8          active s----  5000   22    3328M  0.3%  1d 14h 31m
```

上面的列表显示域处于活动状态，但 s 标志指示域正在停止过程中。此状态应该是短暂状态。

以下示例说明域现在已停止。

```
# ldm list-domain ldg8
NAME          STATE  FLAGS  CONS  VCPU  MEMORY  UTIL  UPTIME
ldg8          bound  -----  5000   22    3328M
```

确定与虚拟网络设备对应的 Solaris 网络接口名称

利用 ldm list-* 命令提供的输出，无法直接确定来宾域上与给定虚拟设备对应的 Solaris OS 网络接口名称。但是，您可以将 ldm list -l 命令输出与 Solaris OS 来宾域的 /devices 下的条目结合使用，来实现此目的。

▼ 查找 Solaris OS 网络接口名称

在此示例中，来宾域 `ldg1` 包含两个虚拟网络设备（`net-a` 和 `net-c`）。要查找 `ldg1` 中与 `net-c` 相对应的 Solaris OS 网络接口名称，请执行以下操作。

1. 使用 `ldm list` 命令查找与 `net-c` 对应的虚拟网络设备实例。

```
# ldm list -l ldg1
...
NETWORK
NAME          SERVICE          DEVICE          MAC
net-a         primary-vsw0@primary  network@0       00:14:4f:f8:91:4f
net-c         primary-vsw0@primary  network@2       00:14:4f:f8:dd:68
...
#
```

与 `net-c` 对应的虚拟网络设备实例为 `network@2`。

2. 要查找 `ldg1` 上相应的网络接口，请登录到 `ldg1`，然后在 `/devices` 下查找该实例的条目。

```
# uname -n
ldg1
# find /devices/virtual-devices@100 -type c -name network@2\*
/devices/virtual-devices@100/channel-devices@200/network@2:vnet1
#
```

网络接口名称是冒号后面条目的一部分；即 `vnet1`。

3. 激活 (**plumb**) `vnet1` 以确定其 **MAC** 地址为 `00:14:4f:f8:dd:68`，并且与步骤 1 的 `ldm list -l` 输出中 `net-c` 对应的 **MAC** 地址相同。

```
# ifconfig vnet1
vnet1: flags=1000842<BROADCAST,RUNNING,MULTICAST,IPv4> mtu 1500 index 3
    inet 0.0.0.0 netmask 0
    ether 0:14:4f:f8:dd:68
#
```

自动或手动分配 MAC 地址

您必须有足够的介质访问控制 (media access control, MAC) 地址，以便分配给计划使用的一定数目的逻辑域、虚拟交换机和虚拟网络。可以让 Logical Domains Manager 自动为逻辑域、虚拟网络 (vnet) 和虚拟交换机 (vswitch) 分配 MAC 地址，也可以从您自己的已分配 MAC 地址池中手动分配 MAC 地址。可设置 MAC 地址的 ldm 子命令包括 add-domain、add-vsw、set-vsw、add-vnet 和 set-vnet。如果您未在这些子命令中指定 MAC 地址，Logical Domains Manager 会自动分配一个地址。

让 Logical Domains Manager 分配 MAC 地址的优势是，它能够利用专用于逻辑域的 MAC 地址块。此外，Logical Domains Manager 可以检测并防止 MAC 地址与同一子网中其他 Logical Domains Manager 实例发生冲突。这样，您就不必手动管理您的 MAC 地址池。

创建逻辑域或将网络设备配置到域中时，即会分配 MAC 地址。此外，除非删除了设备或逻辑域本身，否则此地址分配是持久性的。

本节将介绍下列主题：

- [第 93 页中的“分配给 Logical Domains 软件的 MAC 地址范围”](#)
- [第 94 页中的“自动分配算法”](#)
- [第 94 页中的“检测重复的 MAC 地址”](#)
- [第 95 页中的“释放的 MAC 地址”](#)

分配给 Logical Domains 软件的 MAC 地址范围

已经为逻辑域分配了以下 512 K MAC 地址块：

00:14:4F:F8:00:00 ~ 00:14:4F:FF:FF:FF

Logical Domains Manager 将使用较低的 256 K 地址执行 **MAC 地址自动分配**，因此您不能手动请求以下范围内的地址：

00:14:4F:F8:00:00 - 00:14:4F:FB:FF:FF

您可以使用此范围内的上一半地址执行 **MAC 地址手动分配**：

00:14:4F:FC:00:00 - 00:14:4F:FF:FF:FF

自动分配算法

如果您未在创建逻辑域或网络设备时指定 MAC 地址，Logical Domains Manager 会自动为该逻辑域或网络设备分配 MAC 地址。为了获得此 MAC 地址，Logical Domains Manager 会重复尝试选择地址，然后检查是否存在潜在冲突。

在选择可能的地址之前，Logical Domains Manager 先确定专用于此目的的数据库中是否保存有最近释放的、自动分配的地址（请参见第 95 页中的“[释放的 MAC 地址](#)”）。如果有，Logical Domains Manager 会从数据库中选择该候选地址。

如果没有最近释放的地址，则会从专为此目的留出的 256 K 地址范围中随机选择 MAC 地址。随机选择 MAC 地址可减少重复的 MAC 地址被选作候选地址的几率。

随后将对照其他系统上的其他 Logical Domains Manager 检查所选的地址，以防止实际分配重复的 MAC 地址。所用的算法在第 94 页中的“[检测重复的 MAC 地址](#)”中进行了介绍。如果该地址已经被分配，Logical Domains Manager 将重复操作，选择另一个地址，再检查是否存在冲突。此过程将一直持续下去，直至找到没有分配的 MAC 地址，或超出了 30 秒的时间限制。如果达到了时间限制，则创建设备失败，并显示类似以下内容的错误消息。

Automatic MAC allocation failed. Please set the vnet MAC address manually.

检测重复的 MAC 地址

为避免将同一个 MAC 地址分配给不同的设备，一个 Logical Domains Manager 将对照其他系统上的其他 Logical Domains Manager 进行检查，方法是通过控制域的默认网络接口发送多址广播消息，其中包括 Logical Domains Manager 希望分配给设备的地址。尝试分配 MAC 地址的 Logical Domains Manager 会等待一秒钟的时间，以便返回响应。如果该 MAC 地址已经分配给其他支持 LDom 的系统上的不同设备，则该系统上的 Logical Domains Manager 会返回包含所讨论的 MAC 地址的响应。如果发出请求的 Logical Domains Manager 收到响应，它便知道所选的 MAC 地址已被分配，会选择其他地址，并重复上述操作。

默认情况下，这些多址广播消息仅发送到同一子网中的其他 Logical Domains Manager；默认生存时间 (time-to-live, TTL) 为 1。可以使用服务管理工具 (Service Management Facility, SMF) 属性 `ldmd/hops` 配置 TTL。

每个 Logical Domains Manager 负责：

- 侦听多址广播消息
- 跟踪分配给域的 MAC 地址
- 查找重复项
- 为避免产生重复项而作出响应

如果系统上的 Logical Domains Manager 由于某种原因关闭，则在 Logical Domains Manager 关闭期间可能会产生重复的 MAC 地址。

创建逻辑域或网络设备时会执行自动 MAC 分配，而且自动 MAC 分配会一直保持到该设备或逻辑域被删除。

释放的 MAC 地址

与自动 MAC 地址相关联的逻辑域或设备被删除后，相应的 MAC 地址将保存到最近释放的 MAC 地址的数据库中，供以后在该系统上使用。之所以保存这些 MAC 地址，是为了防止动态主机配置协议 (Dynamic Host Configuration Protocol, DHCP) 服务器的 Internet 协议 (Internet Protocol, IP) 地址耗尽。如果 DHCP 服务器分配 IP 地址，这些服务器会在一段时间（租用时间）内都执行 IP 地址的分配与释放。租用期限通常配置为较长的一段时间，一般是几小时或几天。如果创建和删除网络设备的频率相当高，而 Logical Domains Manager 不自动重用已分配的 MAC 地址，则已分配的 MAC 地址的数量会很快耗尽一个典型配置的 DHCP 服务器所拥有的 MAC 地址数量。

当要求 Logical Domains Manager 自动为逻辑域或网络设备获取 MAC 地址时，它将先查找已释放的 MAC 地址的数据库，以确定是否有先前分配的 MAC 地址可供重用。如果此数据库中具有可用的 MAC 地址，则运行重复 MAC 地址检测算法。如果该 MAC 地址自先前释放后没有分配给任何设备，则将重用该地址，并将其从数据库中删除。如果检测到冲突，则只是从数据库中删除该地址。Logical Domains Manager 随后将尝试数据库中的下一个地址，如果没有可用的地址，将随机选取一个新的 MAC 地址。

CPU 和内存地址映射

Solaris 故障管理体系结构 (Fault Management Architecture, FMA) 可以按物理 CPU 编号报告 CPU 错误，并按物理内存地址报告内存错误。

如果您想要确定发生错误的逻辑域，以及该域中相应的虚拟 CPU 编号或实际内存地址，则必须执行映射。

CPU 映射

可以使用以下过程确定与给定物理 CPU 编号对应的域以及该域中的虚拟 CPU 编号。

▼ 确定 CPU 编号

1. 生成所有域的可解析长列表。

```
primary$ ldm ls -l -p
```

2. 在列表的 VCPU 部分中查找 pid 字段等于物理 CPU 编号的条目。
 - 如果找到了这样的条目，则 CPU 所在的域就是其下方列出此条目的域，该域中的虚拟 CPU 编号由此条目的 vid 字段指定。
 - 如果未找到这样的条目，则 CPU 不在任何域中。

内存映射

可以按以下方式确定与给定物理内存地址 (PA) 对应的域以及该域中的实际内存地址。

▼ 确定实际内存地址

1. 生成所有域的可解析长列表。

```
primary$ ldm ls -l -p
```

2. 在列表的 MEMORY 部分中查找 PA 介于 pa 至 $(pa + size - 1)$ 范围（包含限值，即， $pa \leq PA < (pa + size - 1)$ ）的行。

其中 pa 和 $size$ 是指该行的相应字段的值。

 - 如果找到了这样的条目，则 PA 所在的域就是其下方列出此条目的域，该域中的相应实际地址由 $ra + (PA - pa)$ 指定。
 - 如果未找到这样的条目，则 PA 不在任何域中。

CPU 和内存映射示例

假设您具有如[代码示例 6-16](#)中所示的逻辑域配置，并想要确定与物理 CPU 编号 5 对应的域和虚拟 CPU，以及与物理地址 0x7e816000 对应的域和实际地址。

浏览列表中的 VCPU 条目以找到 pid 字段等于 5 的一个条目，您可以在逻辑域 ldg1 下找到以下条目。

```
|vid=1|pid=5|util=29|strand=100
```

因此，物理 CPU 编号 5 位于域 ldg1 中，在该域中对应的虚拟 CPU 编号为 1。

浏览列表中的 MEMORY 条目，可以在域 ldg2 下找到以下条目。

ra=0x8000000 pa=0x78000000 size=1073741824
--

其中 $0x78000000 \leq 0x7e816000 \leq (0x78000000 + 1073741824 - 1)$ ，即， $pa \leq PA \leq (pa + size - 1)$ 。

因此，PA 位于域 ldg2 中，相应的实际地址为 $0x8000000 + (0x7e816000 - 0x78000000) = 0xe816000$ 。

代码示例 6-16 逻辑域配置的可解析长列表

```
primary$ ldm ls -l -p
VERSION 1.0
DOMAIN|name=primary|state=active|flags=normal,control,vio-service|cons=
SP|ncpu=4|mem=1073741824|util=0.6|uptime=64801|softstate=Solaris running
VCPU
|vid=0|pid=0|util=0.9|strand=100
|vid=1|pid=1|util=0.5|strand=100
|vid=2|pid=2|util=0.6|strand=100
|vid=3|pid=3|util=0.6|strand=100
MEMORY
|ra=0x8000000|pa=0x8000000|size=1073741824
IO
|dev=pci@780|alias=bus_a
|dev=pci@7c0|alias=bus_b
VDS|name=primary-vds0|nclients=2
|vol=disk-ldg1|opts=|dev=/opt/ldoms/testdisk.1
|vol=disk-ldg2|opts=|dev=/opt/ldoms/testdisk.2
VCC|name=primary-vcc0|nclients=2|port-range=5000-5100
VSW|name=primary-vsw0|nclients=2|mac-addr=00:14:4f:fb:42:5c|net-dev=
e1000g0|dev=switch@0|mode=prog,promisc
VCONS|type=SP
DOMAIN|name=ldg1|state=active|flags=normal|cons=5000|ncpu=2|mem=
805306368|util=29|uptime=903|softstate=Solaris running
VCPU
|vid=0|pid=4|util=29|strand=100
|vid=1|pid=5|util=29|strand=100
MEMORY
|ra=0x8000000|pa=0x48000000|size=805306368
VARIABLES
|auto-boot?=true
|boot-device=/virtual-devices@100/channel-devices@200/disk@0
VNET|name=net|dev=network@0|service=primary-vsw0@primary|mac-addr=
00:14:4f:f9:8f:e6
VDISK|name=vdisk-1|vol=disk-ldg1@primary-vds0|dev=disk@0|server=primary
VCONS|group=group1|service=primary-vcc0@primary|port=5000
```

```
DOMAIN|name=ldg2|state=active|flags=normal|cons=5001|ncpu=3|mem=
1073741824|util=35|uptime=775|softstate=Solaris running
VCPU
|vid=0|pid=6|util=35|strand=100
|vid=1|pid=7|util=34|strand=100
|vid=2|pid=8|util=35|strand=100
MEMORY
|ra=0x8000000|pa=0x78000000|size=1073741824
VARIABLES
|auto-boot?=true
|boot-device=/virtual-devices@100/channel-devices@200/disk@0
VNET|name=net|dev=network@0|service=primary-vsw0@primary|mac-addr=
00:14:4f:f9:8f:e7
VDISK|name=vdisk-2|vol=disk-ldg2@primary-vds0|dev=disk@0|server=primary
VCONS|group=group2|service=primary-vc0@primary|port=5000
```

配置分离式 PCI Express 总线以使用多个逻辑域

注 – 对于基于 Sun UltraSPARC T2 的服务器（例如 Sun SPARC Enterprise T5120 和 T5220 服务器），可以为逻辑域分配网络接口单元 (Network Interface Unit, NIU)，而不使用此过程。

基于 Sun UltraSPARC T1 的服务器上的 PCI Express (PCI-E) 总线由两个端口组成，这两个端口连接不同的叶设备。在服务器上分别使用名称 pci@780 (bus_a) 和 pci@7c0 (bus_b) 标识这两个端口。在多域环境中，可以对 PCI-E 总线进行编程，以便使用 Logical Domains Manager 将每个叶分配到不同的域中。这样，您便可以在多个域中直接访问物理设备，而不是使用 I/O 虚拟化。

当 Logical Domains 系统接通电源时，控制 (primary) 域使用所有物理设备资源，因此主域拥有两个 PCI-E 总线叶。



注意 – 在支持的服务器上，所有内部磁盘都连接到一个叶。如果控制域从内部磁盘进行引导，请不要从域中删除该叶。此外，请确保您不会删除具有主网络端口的叶。如果从控制域或服务域中错误地删除了叶，则该域将无法访问所需的设备并变为不可用。如果主网络端口与系统磁盘位于不同的总线上，则将网络电缆移至板载网络端口，并使用 Logical Domains Manager 重新配置虚拟交换机 (vsw) 以反映这一更改。

▼ 创建分离式 PCI 配置

此处显示的示例适用于 Sun Fire T2000 服务器。此外，也可以在其他基于 Sun UltraSPARC T1 的服务器（例如，Sun Fire T1000 服务器和 Netra T2000 服务器）上使用以下过程。虽然面向其他服务器的说明可能与这些说明稍有不同，但是您可以通过此处的示例了解基本原则。主要的一点是，您需要保留具有引导磁盘的叶，从主域中删除另一个叶，并将它分配给另一个域。

1. 检验 primary 域拥有 PCI Express 总线的两个叶。

```
primary# ldm list-bindings primary
...
IO
    DEVICE          PSEUDONYM          OPTIONS
    pci@780         bus_a
    pci@7c0         bus_b
...
```

2. 确定需要保留的引导磁盘的设备路径。

```
primary# df /
/                (/dev/dsk/c1t0d0s0 ): 1309384 blocks  457028 files
```

3. 确定块设备 c1t0d0s0 链接到的物理设备。

```
primary# ls -l /dev/dsk/c1t0d0s0
lrwxrwxrwx  1 root    root          65 Feb  2 17:19 /dev/dsk/c1t0d0s0 -> ../
../devices/pci@7c0/pci@0/pci@1/pci@0,2/LSILogic,sas@2/sd@0,0:a
```

在此示例中，域 primary 的引导磁盘的物理设备位于叶 pci@7c0 下，该叶对应于前面列出的 bus_b。这意味着，我们可以将 PCI-Express 总线的 bus_a (pci@780) 分配给另一个域。

4. 检查 /etc/path_to_inst，以查找板载网络端口的物理路径。

```
primary# grep e1000g /etc/path_to_inst
```

5. 从 primary 域中删除不包含引导磁盘的叶（在此示例中为 pci@780）。

```
primary# ldm remove-io pci@780 primary
```

6. 将此分离式 PCI 配置（在此示例中为 `split-cfg`）添加到系统控制器。

```
primary# ldm add-config split-cfg
```

此外，该配置 (`split-cfg`) 还设置为重新引导后要使用的下一个配置。

注 – 目前，SC 上可以保存的配置数限制为 8 个，不包括 `factory-default` 配置。

7. 重新引导 `primary` 域，以使更改生效。

```
primary# shutdown -i6 -g0 -y
```

8. 将叶（在此示例中为 `pci@780`）添加到需要直接访问的域（在此示例中为 `ldg1`）。

```
primary# ldm add-io pci@780 ldg1
Notice: the LDom Manager is running in configuration mode. Any
configuration changes made will only take effect after the machine
configuration is downloaded to the system controller and the
host is reset.
```

如果您具有 Infiniband 卡，则可能需要在 `pci@780` 总线上启用旁路模式。有关是否需要启用旁路模式的信息，请参见第 101 页中的“在 PCI 总线上启用 I/O MMU 旁路模式”。

9. 重新引导 `ldg1` 域，以使更改生效。

所有域必须处于非活动状态，才能执行此重新引导。如果您是首次配置该域，则该域将处于非活动状态。

```
ldg1# shutdown -i6 -g0 -y
```

10. 确认仍然为 primary 域分配了相应的叶，并为 ldg1 域分配了相应的叶。

primary# ldm list-bindings primary							
NAME	STATE	FLAGS	CONS	VCPU	MEMORY	UTIL	UPTIME
primary	active	-n-cv	SP	4	4G	0.4%	18h 25m
...							
IO							
DEVICE		PSEUDONYM			OPTIONS		
pci@7c0		bus_b					
...							

NAME	STATE	FLAGS	CONS	VCPU	MEMORY	UTIL	UPTIME
ldg1	active	-n---	5000	4	2G	10%	35m
...							
IO							
DEVICE		PSEUDONYM			OPTIONS		
pci@780		bus_a					
...							

此输出确认已将 PCI-E 叶 bus_b 及其下面的设备分配给 primary 域，并将 bus_a 及其设备分配给 ldg1。

在 PCI 总线上启用 I/O MMU 旁路模式

如果具有 Infiniband 主机通道适配器 (Host Channel Adapter, HCA) 卡，则可能需要打开 I/O 内存管理单元 (memory management unit, MMU) 旁路模式。默认情况下，Logical Domains 软件控制 PCI-E 事务，以便给定的 I/O 设备或 PCI-E 选件只能访问在 I/O 域中分配的物理内存。任何访问其他来宾域内存的尝试都将被 I/O MMU 阻止。这在 I/O 域和所有其他域之间提供了更高级别的安全性。但在极少数情况下，PCI-E 或 PCI-X 选件卡无法在 I/O MMU 旁路模式关闭的情况下装入或运行，该选件允许您打开 I/O MMU 旁路模式。不过，如果打开了旁路模式，将不再具有对 I/O 域内存访问的硬件强制保护。

bypass=on 选项可打开 I/O MMU 旁路模式。仅当所有来宾域信任相应的 I/O 域以及该 I/O 域中的 I/O 设备时，才应启用旁路模式。以下示例将打开旁路模式。

```
primary# ldm add-io bypass=on pci@780 ldg1
```

输出显示了 OPTIONS 下的 bypass=on。

使用控制台组

通过虚拟网络终端服务器守护进程 `vntsd(1M)`，您可以使用一个 TCP 端口访问多个域控制台。创建域时，`Logical Domains Manager` 通过为域控制台创建新的默认组，来为每个控制台分配唯一的 TCP 端口。该 TCP 端口随后被分配给控制台组，而非控制台本身。可以使用 `set-vcons` 子命令将控制台绑定到现有的组。

▼ 将多个控制台组成一个组

1. 将域的控制台绑定到一个组。

以下示例显示了将三个不同域（`ldg1`、`ldg2` 和 `ldg3`）的控制台绑定到同一个控制台组（`group1`）。

```
primary# ldm set-vcons group=group1 service=primary-vcc0 ldg1
primary# ldm set-vcons group=group1 service=primary-vcc0 ldg2
primary# ldm set-vcons group=group1 service=primary-vcc0 ldg3
```

2. 连接到相关联的 TCP 端口（在本例中为 `localhost` 上的端口 5000）。

```
# telnet localhost 5000
primary-vnts-group1: h, l, c{id}, n{name}, q:
```

系统会提示您选择一个域控制台。

3. 通过选择 1（列表）列出该组中的域。

```
primary-vnts-group1: h, l, c{id}, n{name}, q: 1
DOMAIN ID          DOMAIN NAME          DOMAIN STATE
0                   ldg1                 online
1                   ldg2                 online
2                   ldg3                 online
```

注 — 要将控制台重新分配到不同的组或 `vcc` 实例，必须将域取消绑定；即，域必须处于非活动状态。有关配置和使用 `SMF` 来管理 `vntsd` 以及使用控制台组的更多信息，请参阅 `Solaris 10 OS vntsd(1M)` 手册页。

将逻辑域从一个服务器移动至另一个服务器

您可以将未运行的逻辑域从一个服务器移动至另一个服务器。在移动域之前，如果在两个服务器上设置了相同的域，则可以更轻松地移动该域。实际上，无需移动域本身；只需将域在一个服务器上取消绑定并停止，然后在另一个服务器上绑定域并启动它。

▼ 设置域以便移动

1. 在两个服务器上创建具有相同名称的域；例如，在 `serverA` 和 `serverB` 上创建 `domainA1`。
2. 向两个服务器添加虚拟磁盘服务器设备和虚拟磁盘。虚拟磁盘服务器会打开底层设备，供导出为绑定的一部分。
3. 仅在一个服务器（例如 `serverA`）上绑定域。使域在另一个服务器上处于非活动状态。

▼ 移动域

1. 在 `serverA` 上取消绑定域并停止它。
2. 在 `serverB` 上绑定域并启动它。

注 – 在绑定域之前，没有使用任何资源。

删除逻辑域

本节介绍了如何删除所有来宾域，并恢复为控制整个服务器的单个 OS 实例。

▼ 删除所有来宾逻辑域

1. 列出系统控制器上的所有逻辑域配置。

```
primary# ldm ls-config
```

2. 删除先前保存到系统控制器 (system controller, SC) 的所有配置 (*config_name*)。对每个这样的配置使用以下命令。

```
primary# ldm rm-config config_name
```

一旦删除了先前保存到 SC 的所有配置，factory-default 域便成为重新引导控制域 (primary) 时要使用的下一个域。

3. 使用 -a 选项停止所有来宾域。

```
primary# ldm stop-domain -a
```

4. 列出所有域，以查看连接到来宾域的所有资源。

```
primary# ldm ls
```

5. 释放连接到来宾域的所有资源。要执行此操作，请对在系统中配置的每个来宾域 (*ldom*) 使用 `ldm unbind-domain` 命令。

注 – 在分离式 PCI 配置中，如果 I/O 域正在提供控制域所需的服务，则可能无法取消绑定 I/O 域。这种情况下，请跳过此步骤。

```
primary# ldm unbind-domain ldom
```

6. 停止控制域。

```
primary# shutdown -il -g0 -y
```

7. 对系统控制器执行关开机循环，以便重新装入 factory-default 配置。

```
sc> poweroff  
sc> poweron
```

在逻辑域中操作 Solaris OS

本节介绍了在实例化 Logical Domains Manager 创建的配置（更确切地说，启用了域）后，使用 Solaris OS 时在行为方面所发生的变化。

注 – 有关是否启用域的任何讨论仅与基于 Sun UltraSPARC T1 的平台有关。否则，将始终启用域。

如果启用了域，则在启动 Solaris OS 之后，OpenBoot 固件不可用

一旦对 Logical Domains Manager 创建的逻辑域配置进行实例化，即可启用域。如果启用了域，在启动 Solaris OS 之后，OpenBoot™ 固件不可用，原因是从内存中删除了该固件。

要想进入 Solaris OS 的 ok 提示符下，您必须停止域。可以使用 Solaris OS 的 halt 命令来停止域。

对服务器执行关开机循环

每当在运行 LDomS 软件的系统上执行需要对服务器进行关开机循环的任何维护操作时，您必须先将当前逻辑域配置保存到 SC。

▼ 将当前逻辑域配置保存到 SC

- 使用以下命令。

```
# ldm add-config config_name
```

OpenBoot power-off 命令的结果

OpenBoot™ power-off 命令无法关闭系统电源。要在处于 OpenBoot 固件中时关闭系统电源，请使用系统控制器或系统处理器的 poweroff 命令。OpenBoot power-off 命令会显示以下消息。

```
NOTICE: power-off command is not supported, use appropriate
NOTICE: command on System Controller to turn power off.
```

在 Solaris OS 中发出中断的结果

如果未启用域，则在发出中断后，Solaris OS 通常会转至 OpenBoot 提示符下。在以下两种情况下会出现本节中所述的行为：

1. 输入设备设置为 keyboard 时按 L1-A 键序。
2. 虚拟控制台处于 telnet 提示符下时输入 send break 命令。

如果启用了域，则在发出这些类型的中断后会出现以下提示。

```
c)ontinue, s)ync, r)eboot, h)alt?
```

键入代表发出这些类型的中断后希望系统执行的操作的字母。

停止或重新引导控制域的结果

下表列出了停止或重新引导控制 (primary) 域的预期行为。

注 – 表 6-1 中有关是否启用域的问题仅与 Sun UltraSPARC T1 处理器有关。否则，将始终启用域。

表 6-1 停止或重新引导控制 (primary) 域的预期行为

命令	是否启用域?	是否配置其他域?	行为
halt	禁用	N/A	对于 Sun UltraSPARC T1 处理器: 转至 ok 提示符。
	启用	否	对于 Sun UltraSPARC T1 处理器: 系统会重置并转至 OpenBoot ok 提示符, 或者转至以下提示符: r)eboot, o)k prompt, or h)alt? 对于 Sun UltraSPARC T2 处理器: 主机断电并保持关机状态, 直到 SC 通电。
	启用	是	如果变量 auto-boot?=true, 则软复位并引导。如果变量 auto-boot?=false, 则软复位并在 ok 提示符处停止。
reboot	禁用	N/A	对于 Sun UltraSPARC T1 处理器: 关闭并再次打开主机电源。
	启用	否	对于 Sun UltraSPARC T1 处理器: 关闭并再次打开主机电源。 对于 Sun UltraSPARC T2 处理器: 重新引导主机, 不关闭电源。
	启用	是	对于 Sun UltraSPARC T1 处理器: 关闭并再次打开主机电源。 对于 Sun UltraSPARC T2 处理器: 重新引导主机, 不关闭电源。
shutdown -i 5	禁用	N/A	对于 Sun UltraSPARC T1 处理器: 关闭主机电源。
	启用	否	主机断电并保持关机状态, 直到 SC 通电。
	启用	是	软复位并重新引导。

将 LDom 与 ALOM CMT 配合使用

本节介绍了将 Advanced Lights Out Manager (ALOM) 芯片多线程 (chip multithreading, CMT) 与 Logical Domains Manager 配合使用时要注意的信息。有关使用 ALOM CMT 软件的更多信息，请参阅《Advanced Lights Out Management (ALOM) CMT v1.3 指南》。



注意 – 由于 ALOM CMT 文档仅提到了一个域，因此您必须清楚 Logical Domains Manager 会引入多个域。如果重新启动了逻辑域，则来宾域的 I/O 服务可能不可用，直到重新启动控制域。这是因为控制域在 Logical Domains Manager 1.0.3 软件中充当服务域。来宾域在重新引导过程中显示为冻结。一旦完全重新启动了控制域，来宾域将恢复正常操作。仅当要关闭整个服务器的电源时，才需要关闭来宾域。

现有的 ALOM CMT 命令可以使用一个附加选项。

```
bootmode [normal|reset_nvram|bootscript=strong|config="config-name"]
```

通过 config="config-name" 选项，您可以将下次打开电源时所用的配置设置为其他配置，包括出厂时提供的 factory-default 配置。

无论主机的电源是打开还是关闭，您都可以调用此命令。它将在下次主机复位或打开电源时生效。

▼ 将逻辑域配置重置为默认配置或其他配置

- 在 ALOM CMT 软件中执行以下命令，将下次打开电源时所用的逻辑域配置重置为出厂默认配置。

```
sc> bootmode config="factory-default"
```

此外，也可以选择已创建（在 Logical Domains Manager 中使用 ldm add-config 命令）的并存储在系统控制器 (system controller, SC) 上的其他配置。执行 ALOM CMT bootmode 命令时，可以使用在 Logical Domains Manager ldm add-config 命令中指定的名称来选择配置。例如，假设您使用名称 ldm-config1 存储了配置。

```
sc> bootmode config="ldm-config1"
```

有关 ldm add-config 命令的更多信息，请参阅 ldm(1M) 手册页或《Logical Domains (LDoms) Manager 1.0.3 Man Page Guide》。

启用并使用 BSM 审计

Logical Domains Manager 使用 Solaris OS 基本安全模块 (Basic Security Module, BSM) 审计功能。BSM 审计功能提供了检查控制域中操作和事件历史记录以确定发生的情况的方法。这些历史记录保存在日志中，其中包括执行的操作、完成时间、执行者以及产生的后果。

如果您准备使用此审计功能，本节介绍了如何启用、检验、禁用、列显输出和轮转审计日志。有关 BSM 审计的详细信息，请参阅 Solaris 10 《系统管理指南：安全性服务》。

可以使用以下两种方法之一启用 BSM 审计。如果要禁用审计，请确保使用启用审计时的同一方法。这两种方法分别是：

- 使用 Solaris Security Toolkit 中的 `enable-bsd.fin` 结束脚本。
默认情况下，`ldm_control-secure.driver` 不使用 `enable-bsd.fin` 脚本。
必须启用选定驱动程序中的结束脚本。
- 使用 Solaris OS `bsmconv(1M)` 命令。

下面介绍了每种方法的操作过程。

▼ 使用 `enable-bsd.fin` 结束脚本

1. 将 `ldm_control-secure.driver` 复制到 `my-ldm.driver`，其中 `my-ldm.driver` 是 `ldm_control-secure.driver` 副本的名称。
2. 将 `ldm_control-config.driver` 复制到 `my-ldm-config.driver`，其中 `my-ldm-config.driver` 是 `ldm_control-config.driver` 副本的名称。
3. 将 `ldm_control-hardening.driver` 复制到 `my-ldm-hardening.driver`，其中 `my-ldm-hardening.driver` 是 `ldm_control-hardening.driver` 副本的名称。
4. 编辑 `my-ldm.driver`，以引用新的配置和强化驱动程序（分别为 `my-ldm-config.driver` 和 `my-ldm-hardening.driver`）。
5. 编辑 `my-ldm-hardening.driver`，并删除驱动程序中以下行前面的井号（#）。

```
enable-bsd.fin
```

6. 执行 `my-ldm.driver`。

```
# /opt/SUNWjass/bin/jass-execute -d my-ldm.driver
```

7. 重新引导 Solaris OS，以使审计生效。

▼ 使用 Solaris OS bsmconv(1M) 命令

1. 在 `/etc/security/audit_control` 文件的 `flags:` 行中添加 `vs`。
2. 运行 `bsmconv(1M)` 命令。

```
# /etc/security/bsmconv
```

有关此命令的更多信息，请参阅 *Solaris 10 Reference Manual Collection* 或手册页。

3. 重新引导 Solaris 操作系统，以使审计生效。

▼ 检验 BSM 审计是否已启用

1. 键入以下命令。

```
# auditconfig -getcond
```

2. 查看输出中是否出现 `audit condition = auditing`。

▼ 禁用审计

可以使用以下两种方法之一禁用审计，具体取决于您启用审计的方法。请参见第 109 页中的“启用并使用 BSM 审计”。

1. 执行以下操作之一。
 - 撤消启用 BSM 审计的 Solaris Security Toolkit 强化操作。

```
# /opt/SUNWjass/bin/jass-execute -u
```

- 使用 Solaris OS `bsmunconv(1M)` 命令。

```
# /etc/security/bsmunconv
```

2. 重新引导 Solaris OS，以使禁用审计生效。

▼ 列显审计输出

- 使用以下方法之一列显 **BSM** 审计输出。
 - 使用 Solaris OS 命令 `auditreduce(1M)` 和 `praudit(1M)` 列显审计输出。例如：

```
# auditreduce -c vs | praudit
# auditreduce -c vs -a 20060502000000 | praudit
```

- 使用 Solaris OS `praudit -x` 命令列显 XML 输出。

▼ 轮转审计日志

- 使用 Solaris OS `audit -n` 命令轮转审计日志。

支持的网络适配器

在逻辑域环境中，服务域中运行的虚拟交换机服务可以直接和与 GLDv3 兼容的网络适配器进行交互。虽然可以在这些系统中使用与 GLDv3 不兼容的网络适配器，但是虚拟交换机不能与这些网络适配器直接进行交互。有关如何使用与 GLDv3 不兼容的网络适配器的信息，请参见第 112 页中的“针对 NAT 和路由配置虚拟交换机和服务域”。

▼ 确定网络适配器是否与 GLDv3 兼容

1. 使用 Solaris OS `dladm(1M)` 命令，示例如下，其中 `bge0` 为网络设备名称。

```
# dladm show-link bge0
bge0                type: non-vlan    mtu: 1500        device: bge0
```

2. 查看输出中的 `type::`
 - 与 GLDv3 兼容的适配器的类型为 `non-vlan` 或 `vlan`。
 - 与 GLDv3 不兼容的适配器的类型为 `legacy`。

针对 NAT 和路由配置虚拟交换机和服务域

虚拟交换机 (vswitch) 是第 2 层交换机，也可用作服务域中的网络设备。可以将虚拟交换机配置为仅充当不同逻辑域中的虚拟网络 (vnet) 设备之间的交换机，但不能通过物理设备与外界网络进行连接。在这种模式下，如果将 vswitch 作为网络设备激活 (plumb) 并启用服务域中的 IP 路由，则虚拟网络可以通过将服务域用作路由器来与外界进行通信。当物理网络适配器不是 GLDv3 兼容的网络适配器时，要实现与域的外部连接，这种操作模式是非常重要的。

此配置的优势如下：

- 虚拟交换机不需要直接使用物理设备，即使底层设备不是 GLDv3 兼容的设备，也能提供外部连接。
- 此配置可以利用 Solaris OS 的 IP 路由和过滤功能。

▼ 设置虚拟交换机以实现与域的外部连接

1. 创建与物理设备毫不相关的虚拟交换机。

如果要分配地址，请确保虚拟交换机具有唯一的 MAC 地址。

```
primary# ldm add-vsw [mac-addr=xxxxxxxxxxxx] primary-vsw0 primary
```

2. 将虚拟交换机作为域正在使用的物理网络设备之外的网络设备激活 (plumb)。

有关激活 (plumb) 虚拟交换机的更多信息，请参见第 43 页中的“将虚拟交换机配置为主接口”。

3. 如果需要，请为虚拟交换机设备配置 DHCP。

有关为虚拟交换机设备配置 DHCP 的更多信息，请参见第 43 页中的“将虚拟交换机配置为主接口”。

4. 如果需要，请创建 /etc/dhcp.vsw 文件。

5. 在服务域中配置 IP 路由，并在所有域中设置所需的路由表。

有关如何执行此操作的信息，请参阅 Solaris Express System Administrator Collection 中的《System Administration Guide: IP Services》的第 5 章“Configuring TCP/IP Network Services and IPv4 Administration”中的“Packet Forwarding and Routing on IPv4 Networks”一节。

在 Logical Domains 环境中配置 IPMP

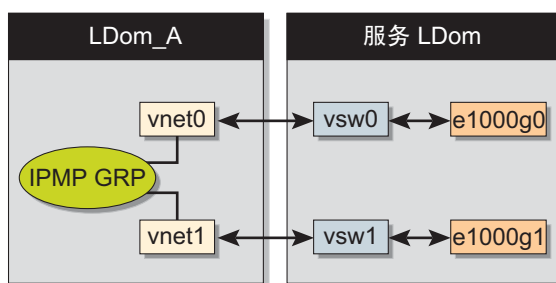
Internet 协议网络多路径 (Internet Protocol Network Multipathing, IPMP) 可跨多个网络接口卡提供容错和负载平衡。通过使用 IPMP, 您可以将一个或多个接口配置到 IP 多路径组中。配置 IPMP 之后, 系统会自动监视 IPMP 组中的接口是否有故障。如果该组中的接口出现故障, 或出于维护需要而被移除, 则 IPMP 会自动迁移或故障转移故障接口的 IP 地址。在 Logical Domains 环境中, 物理接口或虚拟网络接口均可配置为使用 IPMP 进行故障转移。

在逻辑域中将虚拟网络设备配置到 IPMP 组中

通过将逻辑域的虚拟网络设备配置到 IPMP 组中, 可将逻辑域配置为容错。如果设置具有虚拟网络设备的 IPMP 组, 则在活动 - 待机配置中, 请将该组设置为使用基于探测的检测。目前, Logical Domains 1.0.3 软件中的虚拟网络设备不支持基于链路的检测和故障转移。

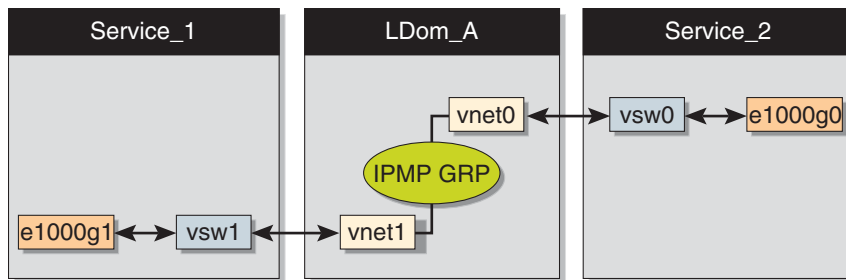
下图显示了两个虚拟网络 (vnet0 和 vnet1), 它们分别连接到服务域中的不同虚拟交换机实例 (vsw0 和 vsw1), 而这两个虚拟交换机又使用两个不同的物理接口 (e1000g0 和 e1000g1)。如果一个物理接口出现故障, LDom_A 中的 IP 层可通过基于探测的检测, 检测到该故障, 并失去相应的 vnet 上的连接, 然后自动故障转移到辅助 vnet 设备。

图 6-1 两个连接到不同虚拟交换机实例的虚拟网络



在逻辑域中, 通过将每个虚拟网络设备 (vnet0 和 vnet1) 连接到不同服务域中的虚拟交换机实例 (如下图所示), 可获得更高的可靠性。可以使用分离式 PCI 配置, 对两个带有虚拟交换机实例 (vsw1 和 vsw2) 的服务域 (Service_1 和 Service_2) 进行设置。这种情况下, 除了网络硬件故障外, LDom_A 可以检测到虚拟网络故障, 并在服务域崩溃或关闭之前触发故障转移。

图 6-2 连接到不同服务域的各个虚拟网络设备



有关如何配置和使用 IPMP 组的更多信息，请参阅 Solaris 10 《系统管理指南：IP 服务》。

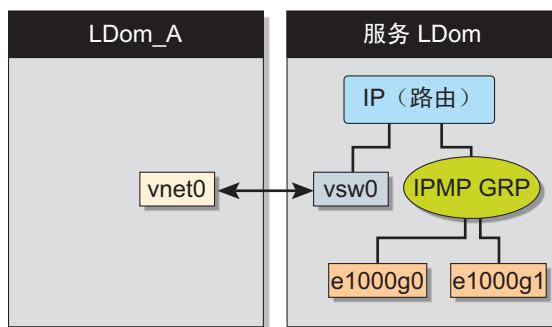
在服务域中配置并使用 IPMP

在 Logical Domains 环境中，通过在服务域中将物理接口配置到 IPMP 组中，也可以设置网络故障检测和恢复。要进行此操作，请在服务域中将虚拟交换机配置为网络设备，并将服务域本身配置为充当 IP 路由器。（有关设置 IP 路由的信息，请参阅 Solaris 10 《系统管理指南：IP 服务》。）

进行配置后，虚拟交换机将所有来自虚拟网络（目标是外部机器）的包发送到其 IP 层，而不是直接通过物理设备发送包。如果物理接口出现故障，IP 层会检测故障，并自动通过辅助接口对包进行重新路由。

由于物理接口被直接配置到 IPMP 组中，因此可以将该组设置为使用基于链路的检测或基于探测的检测。下图显示了两个配置为属于 IPMP 组的网络接口（e1000g0 和 e1000g1）。虚拟交换机实例 (vsw0) 已作为网络设备激活 (plumb)，以便向其 IP 层发送包。

图 6-3 两个配置为属于 IPMP 组的网络接口



词汇表

本列表定义了 Logical Domains 1.0.3 文档中的术语、缩写和首字母缩略词。

A

ALOM CMT	Advanced Lights Out Manager chip multithreading（Advanced Lights Out Manager 芯片多线程）；它运行在系统控制器上，可用于监视和控制 CMT 服务器
auditreduce(1M)	合并和选择审计跟踪文件中的审计记录
auditing （审计）	使用 Solaris OS BSM 识别安全更改源
authorization （授权）	使用 Solaris OS RBAC 设置授权

B

bge	用于 Broadcom BCM57xx 设备的 Broadcom 千兆位以太网驱动程序
BSM	Basic Security Module（基本安全模块）
bsmconv(1M)	启用 BSM
bsmunconv(1M)	禁用 BSM

C

- CD** compact disc (光盘)
- CLI** command-line interface (命令行界面)
- compliance**
(遵从性) 确定系统配置是否符合预先定义的安全配置文件
- config** 保存在系统控制器上的逻辑域配置的名称
- CMT** chip multithreading (芯片多线程)
- constraints (约束)** 对于 Logical Domains Manager, 约束是您希望分配给特定域的一个或多个资源。您会接收到要求添加到域中的所有资源, 或者任何资源也得不到, 这取决于可用资源。
- control domain**
(控制域) 创建和管理其他逻辑域及服务的域
- CPU** central processing unit (中央处理器)
- CWQ** Control Word Queue (控制字队列); 用于基于 Sun UltraSPARC T2 的平台的加密单元

D

- DHCP** Dynamic Host Configuration Protocol (动态主机配置协议)
- DMP** Dynamic Multipathing (动态多路径) (Veritas)
- DR** dynamic reconfiguration (动态重新配置)
- drd(1M)** 用于 Logical Domains Manager (Solaris 10 OS) 的动态重新配置守护进程
- DS** Domain Service (域服务) 模块 (Solaris 10 OS)
- DVD** digital versatile disc (数字通用光盘)

E

- e1000g** 用于 Intel PRO/1000 千兆位系列网络接口控制器的驱动程序
- EFI** extensible firmware interface （可扩展固件接口）
- ETM** Encoding Table Management （编码表管理）模块 (Solaris 10 OS)

F

- FC_AL** Fiber Channel Arbitrated Loop （光纤通道仲裁环路）
- FMA** Fault Management Architecture （故障管理体系结构）
- fmd(1M)** Fault Manager 守护进程 (Solaris 10 OS)
- format(1M)** 磁盘分区和维护实用程序
- fmthard(1M)** 填充硬盘的标签
- FTP** File Transfer Protocol （文件传输协议）

G

- guest domain**
（来宾域） 它使用来自 I/O 域和服务域的服务，并由控制域进行管理。
- GLDv3** Generic LAN Driver version 3 （Generic LAN Driver 版本 3）。

H

- hardening** （强化） 修改 Solaris OS 配置以提高安全性
- HDD** hard disk drive （硬盘驱动器）
- hypervisor** （虚拟机
管理程序） 在操作系统和硬件层之间插入的固件层

I

- `io` I/O 设备，例如内部磁盘和 PCI-Express (PCI-E) 控制器及其连接的适配器和设备
- IB** Infiniband
- IDE** integrated development environment（集成开发环境）
- I/O domain（I/O 域）** 此域对物理 I/O 设备具有直接拥有权和直接访问权，而且它与其他逻辑域共享以虚拟设备形式出现的那些设备
- `ioctl` input/output control call（输入/输出控制调用）
- IP** Internet Protocol（Internet 协议）
- IPMP** Internet Protocol Network Multipathing（Internet 协议网络多路径）
- ISO** International Organization for Standardization（国际标准化组织）

K

- `kaio` kernel asynchronous input/output（内核异步输入/输出）
- KB** kilobyte（千字节）
- KU** kernel update（内核更新）

L

- LAN** local-area network（局域网）
- LDAP** Lightweight Directory Access Protocol（轻量目录访问协议）
- LDC** logical domain channel（逻辑域通道）
- ldm(1M)** Logical Domains Manager 实用程序
- `ldmd` Logical Domains Manager daemon（Logical Domains Manager 守护进程）
- `lofi` loopback file（回送文件）

logical domain (逻辑域)	在单个计算机系统内具有自己的操作系统、资源和身份的离散逻辑分组
Logical Domains (LDoms) Manager	可提供 CLI 以创建和管理逻辑域，并将资源分配给域
LUN	logical unit number (逻辑单元号)

M

MAC	介质访问控制 (media access control, MAC) 地址; LDoms 可以自动分配 MAC 地址, 您也可以手动分配 MAC 地址
MAU	Modular Arithmetic Unit (模运算单元); 用于基于 Sun UltraSPARC T1 的平台 的加密设备
MB	megabyte (兆字节)
MD	服务器数据库中的机器描述 (machine description, MD)
mem, memory	内存单元默认大小 (以字节为单位), 或者指定千兆字节 (G)、千字节 (K) 或兆字 节 (M)。可分配给来宾域的服务器虚拟内存。
metadb(1M)	创建和删除 SVM 元设备状态数据库的副本
metaset(1M)	配置磁盘集
mhd(7I)	多主机磁盘控制操作
MMF	Multimode fiber (多模光纤)
MIB	Management Information Base (管理信息库)
minimizing (最小化)	安装所必需的最少数量的核心 Solaris OS 软件包
MMU	memory management unit (内存管理单元)
mtu	maximum transmission unit (最大传输单元)

N

NAT	Network Address Translation（网络地址转换）
NDPSS	Netra Data Plane Software Suite
ndpsldcc	Netra Data Plane Software Logical Domain Channel Client（Netra Data Plane 软件逻辑域通道客户机）。另请参见 vdpccl 。
ndpsldcs	Netra Data Plane Software Logical Domain Channel Service（Netra Data Plane 软件逻辑域通道服务）。另请参见 vdpcs 。
NFS	Network File System（网络文件系统）
NIS	Network Information Service（网络信息服务）
NIU	Network Interface Unit（网络接口单元）（Sun SPARC Enterprise T5120 和 T5220 服务器）
NTS	network terminal server（网络终端服务器）
NVRAM	non-volatile random-access memory（非易失性随机存取存储器）
nxge	用于 Sun x8 Express 1/10G 以太网适配器的驱动程序

O

OS	operating system（操作系统）
-----------	------------------------

P

PA	physical address（物理地址）
PCI	peripheral component interconnect（外设部件互连）总线
PCI-E	PCI Express (PCI-E) 总线
PCI-X	PCI Extended (PCI-X) 总线
PICL	Platform Information and Control Library（平台信息和控制库）
picld(1M)	PICL 守护进程

praudit(1M) 列显审计跟踪文件的内容

PRI priority (优先级)

R

RA real address (实际地址)

RAID Redundant Array of Inexpensive Disk (廉价磁盘冗余阵列)

RBAC Role-Based Access Control (基于角色的访问控制)

RPC Remote Procedure Call (远程过程调用)

S

SC system controller (系统控制器); 同系统处理器

SCSI Small Computer System Interface (小型计算机系统接口)

service domain

(服务域) 向其他逻辑域提供诸如虚拟交换机、虚拟控制台连接器和虚拟磁盘服务器等设备的逻辑域

SMA System Management Agent (系统管理代理)

SMF Solaris 10 OS 的服务管理工具 (Service Management Facility, SMF)

SNMP Simple Network Management Protocol (简单网络管理协议)

SP system processor (系统处理器); 同系统控制器

SSH Secure Shell (安全 Shell)

ssh(1) 安全 Shell 命令

sshd(1M) 安全 Shell 守护进程

SunVTS Sun Validation Test Suite (Sun 验证测试套件)

svcadm(1M) 处理服务实例

SVM Solaris Volume Manager (Solaris 卷管理器)

T

TCP Transmission Control Protocol （传输控制协议）

U

UDP User Datagram Protocol （用户图表协议）

USB Universal Serial Bus （通用串行总线）

uscsi(7D) 用户 SCSI 命令界面

UTP unshielded twisted pair （非屏蔽双绞线）

V

vBSC virtual blade system controller （虚拟刀片系统控制器）

vcc, vconscon 虚拟控制台集中器服务，带有分配到来宾域的特定端口范围

vcons, vconsole 用于访问系统级消息的虚拟控制台。通过连接到控制域中特定端口上的 vconscon 服务来实现连接。

vcpu virtual central processing unit （虚拟中央处理器）。服务器的每个内核都能够以若干个虚拟 CPU 来表示。例如，一个 8 核的 Sun Fire T2000 服务器具有可以在逻辑域之间分配的 32 个虚拟 CPU。

vdc virtual disk client （虚拟磁盘客户机）

vdpc NDPS 环境中的虚拟数据平面通道客户机 (virtual data plane channel client, vdpc)

vdpcs NDPS 环境中的虚拟数据平面通道服务 (virtual data plane channel service, vdpcs)

vdisk 虚拟磁盘是各种类型的物理设备、卷或文件所支持的通用块设备。

vds, vdiskserver 通过虚拟磁盘服务器可以将虚拟磁盘导入到逻辑域中。

vdsdev, vdiskserverdevice 虚拟磁盘服务器设备是由虚拟磁盘服务器导出的。该设备可以是整个磁盘、一个磁盘片、一个文件或一个磁盘卷。

vnet	虚拟网络设备可实现虚拟以太网设备，还可以使用虚拟网络交换机 (vswitch) 与系统中的其他 vnet 设备进行通信。
vntsd(1M)	Logical Domains 控制台 (Solaris 10 OS) 的虚拟网络终端服务器守护进程
volfs(7FS)	卷管理文件系统
vsw, vswitch	虚拟网络交换机，它可将虚拟网络设备连接到外部网络，还可以在虚拟网络设备和外部网络之间交换包。
VTOC	volume table of contents （卷目录）
VxDMP	Veritas Dynamic Multipathing （Veritas 动态多路径）
VxVM	Veritas Volume Manager （Veritas 卷管理器）

W

WAN	wide-area network （广域网）
------------	-------------------------

X

XFP	eXtreme Fast Path （极速路径）
XML	Extensible Markup Language （可扩展标记语言）

Z

ZFS	Zettabyte File System （Zettabyte 文件系统） (Solaris 10 OS)
zpool(1M)	ZFS 存储池
ZVOL	ZFS 卷仿真驱动程序

