



Solaris Volume Manager 管理指南



Sun Microsystems, Inc.
4150 Network Circle
Santa Clara, CA 95054
U.S.A.

文件号码 819-7064-10
2006 年 10 月

版权所有 2006 Sun Microsystems, Inc. 4150 Network Circle, Santa Clara, CA 95054 U.S.A. 保留所有权利。

本文档及其相关产品的使用、复制、分发和反编译均受许可证限制。未经 Sun 及其许可方（如果有）的事先书面许可，不得以任何形式、任何手段复制本产品或文档的任何部分。第三方软件，包括字体技术，均已从 Sun 供应商处获得版权和使用许可。

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、docs.sun.com、AnswerBook、AnswerBook2、和 Solaris 是 Sun Microsystems, Inc. 在美国和其他国家/地区的商标或注册商标。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。

OPEN LOOK 和 SunTM 图形用户界面是 Sun Microsystems, Inc. 为其用户和许可证持有者开发的。Sun 感谢 Xerox 在研究和开发可视或图形用户界面的概念方面为计算机行业所做的开拓性贡献。Sun 已从 Xerox 获得了对 Xerox 图形用户界面的非独占性许可证，该许可证还适用于实现 OPEN LOOK GUI 和在其他方面遵守 Sun 书面许可协议的 Sun 许可证持有者。

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。

目录

前言	19
1 Solaris Volume Manager 入门	23
Solaris Volume Manager 汇总信息—新增功能	24
Solaris Volume Manager 汇总信息—存储容量	24
Solaris Volume Manager 汇总信息—可用性	25
Solaris Volume Manager 汇总信息—I/O 性能	25
Solaris Volume Manager 汇总信息—管理	26
Solaris Volume Manager 汇总信息—故障排除	27
2 存储管理概念	29
存储管理介绍	29
存储硬件	29
RAID 级别	30
配置规划指南	30
选择存储设备	31
常规性能指南	32
随机 I/O 和顺序 I/O 优化	33
随机 I/O	33
顺序访问 I/O	33
3 Solaris Volume Manager 概述	35
Solaris Volume Manager 中的新增功能	35
Solaris Volume Manager 介绍	35
Solaris Volume Manager 如何管理存储	36
如何管理 Solaris Volume Manager	36
▼ 如何访问 Solaris Volume Manager 图形用户界面 (Graphical User Interface, GUI)	37

Solaris Volume Manager 要求	38
Solaris Volume Manager 组件概述	38
卷概述	39
状态数据库和状态数据库副本	42
热备用池	43
磁盘集	43
Solaris Volume Manager 配置原则	44
一般原则	44
文件系统挂载原则	44
有关创建 Solaris Volume Manager 组件的概述	44
创建 Solaris Volume Manager 组件的先决条件	45
Solaris Volume Manager 中的多 TB 支持的概述	45
大卷支持限制	45
使用大卷	46
升级到 Solaris Volume Manager	46
4 Solaris Volume Manager for Sun Cluster (概述)	49
Solaris Volume Manager for Sun Cluster 介绍	49
先决条件：多属主磁盘集功能所需的软件组件	50
多属主磁盘集概念	51
与多属主磁盘集关联的任务	53
Solaris Volume Manager for Sun Cluster 配置	54
多属主磁盘集中的 RAID-1 (镜像) 卷	55
多属主磁盘集的镜像拥有权	55
数据管理和恢复过程	56
5 配置和使用 Solaris Volume Manager (方案)	59
方案背景信息	59
硬件配置	59
初始物理存储配置	60
最终的 Solaris Volume Manager 配置	60
6 状态数据库 (概述)	63
关于 Solaris Volume Manager 状态数据库和副本	63

了解多数一致算法	64
管理状态数据库副本	65
处理状态数据库副本错误	66
方案一状态数据库副本	67
7 状态数据库（任务）	69
状态数据库副本（任务列表）	69
创建状态数据库副本	70
▼如何创建状态数据库副本	70
维护状态数据库副本	72
▼如何检查状态数据库副本的状态	72
▼如何删除状态数据库副本	73
8 RAID-0（条带化和串联）卷（概述）	75
RAID-0 卷概述	75
RAID-0（条带化）卷	76
RAID-0（串联）卷	77
RAID-0（串联条带化）卷	79
创建 RAID-0 卷的背景信息	81
RAID-0 卷要求	81
RAID-0 卷指南	81
方案一RAID-0 卷	81
9 RAID-0（条带化和串联）卷（任务）	83
RAID-0 卷（任务列表）	83
创建 RAID-0（条带化）卷	83
▼如何创建 RAID-0（条带化）卷	84
创建 RAID-0（串联）卷	85
▼如何创建 RAID-0（串联）卷	85
扩展存储容量	86
▼如何扩展现有数据的存储容量	87
▼如何扩展现有的 RAID-0 卷	88
删除 RAID-0 卷	90
▼如何删除 RAID-0 卷	90

10 RAID-1 (镜像) 卷 (概述)	93
RAID-1 (镜像) 卷概述	93
子镜像概述	93
方案一 RAID-1 (镜像) 卷	94
提供 RAID-1+0 和 RAID-0+1	94
RAID-1 卷 (镜像) 重新同步	96
完整重新同步	96
优化的重新同步	96
部分重新同步	96
创建和维护 RAID-1 卷	97
RAID-1 卷的配置指南	97
RAID-1 卷的性能指南	98
关于 RAID-1 卷选项	98
了解子镜像状态以确定维护操作	99
引导到单用户模式对 RAID-1 卷的影响	101
方案一 RAID-1 卷 (镜像)	101
11 RAID-1 (镜像) 卷 (任务)	103
RAID-1 卷 (任务列表)	103
创建 RAID-1 卷	104
▼ 如何从未使用的片创建 RAID-1 卷	104
▼ 如何从文件系统创建 RAID-1 卷	107
▼ SPARC: 如何从根 (/) 文件系统创建 RAID-1 卷	112
x86: 从根 (/) 文件系统创建 RAID-1 卷	116
▼ x86: 如何使用 GRUB 从根 (/) 文件系统创建 RAID-1 卷	117
▼ x86: 如何使用 DCA 从根 (/) 文件系统创建 RAID-1 卷	121
了解对根 (/) 文件系统创建镜像时的引导时警告	127
处理子镜像	128
▼ 如何附加子镜像	128
▼ 如何拆离子镜像	130
▼ 如何使子镜像脱机和联机	130
▼ 如何启用子镜像中的片	132
维护 RAID-1 卷	132
▼ 如何查看镜像和子镜像的状态	132
▼ 如何更改 RAID-1 卷选项	136

▼ 如何扩展 RAID-1 卷	137
响应 RAID-1 卷组件的故障	138
▼ 如何替换子镜像中的片	138
▼ 如何替换子镜像	140
删除 RAID-1 卷（取消镜像）	142
▼ 如何取消镜像文件系统	142
▼ 如何取消镜像无法挂载的文件系统	145
在 RAID-1 卷上备份数据	149
▼ 如何对 RAID-1 卷执行联机备份	149
12 软分区（概述）	153
软分区概述	153
软分区的配置指南	154
方案—软分区	154
13 软分区（任务）	155
软分区（任务列表）	155
创建软分区	155
▼ 如何创建软分区	155
维护软分区	157
▼ 如何检查软分区的状态	157
▼ 如何扩展软分区	158
▼ 如何删除软分区	159
14 RAID-5 卷（概述）	161
RAID-5 卷的概述	161
示例—RAID-5 卷	162
示例—串联（扩展）RAID-5 卷	162
创建 RAID-5 卷的背景信息	164
RAID-5 卷的要求	164
RAID-5 卷指南	164
有关检查 RAID-5 卷状态的概述	165
有关替换和启用 RAID-5 卷中的片的概述	167
方案—RAID-5 卷	167

15 RAID-5 卷（任务）	169
RAID-5 卷（任务列表）	169
创建 RAID-5 卷	170
▼ 如何创建 RAID-5 卷	170
维护 RAID-5 卷	171
▼ 如何检查 RAID-5 卷的状态	171
▼ 如何扩展 RAID-5 卷	172
▼ 如何在 RAID-5 卷中启用组件	174
▼ 如何在 RAID-5 卷中替换组件	174
16 热备用池（概述）	179
热备件和热备用池的概述	179
热备件	179
热备用池	180
热备件的工作原理	180
热备用池状态	181
示例—热备用池	181
方案—热备件	182
17 热备用池（任务）	183
热备用池（任务列表）	183
创建热备用池	184
▼ 如何创建热备用池	184
▼ 如何向热备用池中添加更多的片	185
将热备用池与卷相关联	186
▼ 如何将热备用池与卷相关联	186
▼ 如何更改相关的热备用池	188
维护热备用池	190
▼ 如何检查热备件和热备用池的状态	190
▼ 如何替换热备用池中的热备件	191
▼ 如何从热备用池中删除热备件	192
▼ 如何启用热备件	193

18 磁盘集（概述）	195
磁盘集方面的新增功能	195
磁盘集介绍	195
磁盘集类型	196
本地磁盘集	196
已命名的磁盘集	196
Solaris Volume Manager 磁盘集管理	198
保留磁盘集	198
释放磁盘集	199
导入磁盘集	199
自动磁盘分区	200
磁盘集名称要求	203
示例—两个共享磁盘集	203
磁盘集使用指南	204
磁盘集中的异步共享存储	205
方案—磁盘集	205
19 磁盘集（任务）	207
磁盘集（任务列表）	207
创建磁盘集	208
▼ 如何创建磁盘集	208
扩展磁盘集	210
▼ 如何向磁盘集内添加磁盘	210
▼ 如何向磁盘集内添加另一台主机	211
▼ 如何在磁盘集内创建 Solaris Volume Manager 组件	212
维护磁盘集	215
▼ 如何检查磁盘集的状态	215
▼ 如何从磁盘集内删除磁盘	216
▼ 如何提取磁盘集	217
▼ 如何释放磁盘集	219
▼ 如何删除主机或磁盘集	220
导入磁盘集	222
▼ 如何列显有关可导入的磁盘集的报告	223
▼ 如何将磁盘集从一个系统导入到另一个系统	224

20 维护 Solaris Volume Manager (任务)	227
Solaris Volume Manager 维护 (任务列表)	227
查看 Solaris Volume Manager 配置	228
▼ 如何查看 Solaris Volume Manager 卷配置	228
其他参考信息	235
重命名卷	235
重命名卷的背景信息	235
交换卷名称	236
▼ 如何重命名卷	236
使用配置文件	238
▼ 如何创建配置文件	238
▼ 如何从配置文件中初始化 Solaris Volume Manager	238
更改 Solaris Volume Manager 缺省值	240
使用 growfs 命令扩展文件系统	240
扩展片和卷的背景信息	241
▼ 如何扩展文件系统	241
在 RAID-1 和 RAID-5 卷中替换和启用组件的概述	242
启用组件	243
使用其他可用组件来替换组件	243
Maintenance (维护) 和 Last Erred (最近出错) 状态	244
在 RAID-1 和 RAID-5 卷中替换和启用组件的背景信息	245
21 Solaris Volume Manager 的最佳做法	247
部署小型服务器	247
结合使用 Solaris Volume Manager 和网络存储设备	249
22 自上而下创建卷 (概述)	251
自上而下创建卷概述	251
使用磁盘集实现自上而下创建卷	252
自上而下创建卷的过程	252
确定可用于自上而下创建卷的磁盘	254
23 自上而下创建卷 (任务)	255
自上而下创建卷 (任务列表)	255

自上而下创建卷的先决条件	256
自动创建卷	256
通过指定输出的详细程度来分析卷的创建过程	257
▼ 如何使用 metassist 命令来创建 RAID-1（镜像）卷	257
使用 metassist 命令处理基于文件的数据	261
使用 metassist 命令创建命令文件（Shell 脚本）	261
▼ 如何使用 metassist 命令来创建命令文件（Shell 脚本）	261
使用保存的 Shell 脚本（由 metassist 命令创建）创建卷	268
▼ 如何执行保存的 metassist 命令 Shell 脚本	268
使用 metassist 命令创建卷配置文件	269
▼ 如何使用 metassist 命令来创建卷配置文件	269
更改 metassist 命令的缺省行为	272
更改卷缺省配置文件	272
24 监视和错误报告（任务）	275
Solaris Volume Manager 监视和报告（任务列表）	275
配置 mdmonitord 命令以定期检查错误	276
▼ 如何配置 mdmonitord 命令以定期检查错误	276
Solaris Volume Manager SNMP 代理概述	277
配置 Solaris Volume Manager SNMP 代理	278
▼ 如何配置 Solaris Volume Manager SNMP 代理	278
Solaris Volume Manager SNMP 代理的限制	281
用 cron 作业监视 Solaris Volume Manager	281
▼ 如何自动检查卷中的错误	281
25 对 Solaris Volume Manager 进行疑难解答（任务）	291
对 Solaris Volume Manager 进行疑难解答（任务列表）	291
系统疑难解答概述	292
对系统进行疑难解答的先决条件	292
对 Solaris Volume Manager 进行疑难解答的一般原则	292
一般疑难解答方法	293
替换磁盘	293
▼ 如何替换出现故障的磁盘	293
从磁盘移动问题中恢复	296
磁盘移动和设备 ID 概述	296

解决未命名设备错误	296
升级到 Solaris 10 发行版之后设备 ID 的改变	297
从引导问题中恢复	299
有关引导问题的背景信息	300
如何从错误的 /etc/vfstab 项中恢复	300
▼ 从根 (/) RAID-1（镜像）卷中恢复	300
▼ 如何从引导设备故障中恢复	303
从状态数据库副本故障中恢复	310
▼ 如何从状态数据库副本不足中恢复	310
从软分区问题中恢复	314
▼ 如何恢复软分区的配置数据	314
从另一个系统中恢复存储	317
▼ 如何从本地磁盘集恢复存储	317
从已知的磁盘集恢复存储	326
▼ 如何列显有关可导入的磁盘集的报告	326
▼ 如何将磁盘集从一个系统导入到另一个系统	327
从磁盘集问题中恢复	329
无法获取磁盘集拥有权时应执行的操作	329
▼ 如何清除磁盘集	329
使用 ufsdump 命令对已挂载的文件系统执行备份	331
▼ 如何针对 RAID-1 卷上的已挂载文件系统执行备份	331
执行系统恢复	333
▼ 如何使用 Solaris Volume Manager 配置来恢复系统	333
A 重要的 Solaris Volume Manager 文件	335
系统文件和启动文件	335
手动配置的文件	336
md.tab 文件概述	336
B Solaris Volume Manager 快速参考	339
命令行参考	339
C Solaris Volume Manager CIM/WBEM API	341
管理 Solaris Volume Manager	341

索引	343
----------	-----

表

表 2-1	存储类型的比较	31
表 2-2	优化冗余存储	31
表 3-1	Solaris Volume Manager 功能汇总	38
表 3-2	卷的类别	39
表 10-1	RAID-1 卷读取策略	99
表 10-2	RAID-1 卷写入策略	99
表 10-3	子镜像状态	100
表 10-4	子镜像片状态	100
表 14-1	RAID-5 卷状态	165
表 14-2	RAID-5 片状态	166
表 16-1	热备用池状态（命令行）	181
表 18-1	磁盘集的示例卷名称	203
表 25-1	Solaris Volume Manager 的常见引导问题	300
表 B-1	Solaris Volume Manager 命令	339



图 3-1	Solaris Management Console 中增强的存储工具 (Solaris Volume Manager) 视图	37
图 3-2	卷、物理磁盘和片之间的关系	41
图 4-1	群集配置样例	50
图 5-1	基本硬件存储方案图	60
图 8-1	RAID-0（条带化）卷示例	77
图 8-2	RAID-0（串联）卷示例	79
图 8-3	RAID-0（串联条带化）卷示例	80
图 10-1	RAID-1（镜像）示例	94
图 10-2	RAID-1+0 示例	95
图 14-1	RAID-5 卷示例	162
图 14-2	扩展的 RAID-5 卷示例	163
图 16-1	热备用池示例	182
图 18-1	磁盘集示例	204
图 21-1	小型系统配置	248
图 22-1	用于自上而下创建卷的处理选项	253

前言

《Solaris Volume Manager 管理指南》介绍如何使用 Solaris™ Volume Manager 来管理系统的存储需求。使用 Solaris Volume Manager，可以创建、修改和使用 RAID-0（串联和条带化）卷、RAID-1（镜像）卷、RAID-5 卷和软分区。

注 - 此 Solaris 发行版支持使用以下 SPARC® 和 x86 系列处理器体系结构的系统：UltraSPARC®、SPARC64、AMD64、Pentium 和 Xeon EM64T。支持的系统可以在 <http://www.sun.com/bigadmin/hcl> 上的《Solaris 10 Hardware Compatibility List》中找到。本文档列举了在不同类型的平台上进行实现时的所有差别。

在本文档中，这些与 x86 相关的术语表示以下含义：

- x86 泛指 64 位和 32 位的 x86 兼容产品系列。
- x64 指出了有关 AMD64 或 EM64T 系统的特定 64 位信息。
- 32 位 x86 指出了有关基于 x86 的系统的特定 32 位信息。

若想了解本发行版支持哪些系统，请参见《Solaris 10 Hardware Compatibility List》。

目标读者

系统和存储管理员可以使用本书确定以下各项：

- Solaris Volume Manager 支持的任务
- 使用 Solaris Volume Manager 提供更可靠且更易访问数据的方法

本书的结构

《Solaris Volume Manager 管理指南》包含以下信息：

第 1 章提供本书中介绍的概念和任务的详细“汇总信息”。使用本章作为本书内容的导航帮助。

第 2 章向初学此技术的读者介绍一般存储管理概念。

第 3 章介绍 Solaris Volume Manager。本章介绍与产品相关的基本概念并说明如何访问 Solaris Volume Manager 工具。

第 4 章介绍多属主磁盘集。多属主磁盘集增强了 Solaris Volume Manager 在 Sun™ Cluster 环境中的用途。

第 5 章提供本书中所使用的存储配置方案。此方案用于帮助您了解 Solaris Volume Manager 产品。

第 6 章介绍与状态数据库和状态数据库副本相关的概念。

第 7 章说明如何执行与状态数据库和状态数据库副本相关的任务。

第 8 章介绍与 RAID-0（条带化和串联）卷相关的概念。

第 9 章说明如何执行与 RAID-0（条带化和串联）卷相关的任务。

第 10 章介绍与 RAID-1（镜像）卷相关的概念。

第 11 章说明如何执行与 RAID-1（镜像）卷相关的任务。

第 12 章介绍与 Solaris Volume Manager 的软分区功能相关的概念。

第 13 章说明如何执行与软分区相关的任务。

第 14 章介绍与 RAID-5 卷相关的概念。

第 15 章说明如何执行与 RAID-5 卷相关的任务。

第 16 章介绍与热备件和热备用池相关的概念。

第 17 章说明如何执行与热备件和热备用池相关的任务。

第 18 章介绍与磁盘集相关的概念。

第 19 章说明如何执行与磁盘集相关的任务。

第 20 章说明一些与特定 Solaris Volume Manager 组件无关的一般维护任务。

第 21 章提供有关配置和使用 Solaris Volume Manager 的一些“最佳做法”信息。

第 23 章介绍与 Solaris Volume Manager 自上而下卷创建功能相关的概念和任务。

第 24 章提供有关使用 Solaris Volume Manager SNMP 代理和其他错误检查方法的概念和说明。

第 25 章提供有关排除和解决 Solaris Volume Manager 环境下常见问题的信息。

附录 A 列出了重要的 Solaris Volume Manager 文件。

附录 B 提供了包含命令和其他有帮助信息的汇总表。

附录 C 简要介绍了允许通过与 WBEM 兼容的管理工具来管理开放式 Solaris Volume Manager 的 CIM/WBEM API。

相关书籍

Solaris Volume Manager 是可用于 Solaris 操作系统的几个系统管理工具之一。下面提供了有关总体系统管理特征和功能以及相关工具的信息：

- 《系统管理指南：基本管理》
- 《系统管理指南：高级管理》
- 《系统管理指南：设备和文件系统》

文档、支持和培训

Sun 提供的服务	URL	说明
文档	http://www.sun.com/documentation/	下载 PDF 及 HTML 格式的文档，购买印刷文档
支持和培训	http://www.sun.com/supporttraining/	获取技术支持、下载修补程序，以及学习 Sun 提供的课程

印刷约定

下表介绍了本书中的印刷约定。

表 P-1 印刷约定

字体或符号	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 <code>ls -a</code> 列出所有文件。 machine_name% you have mail.
AaBbCc123	用户键入的内容，与计算机屏幕输出的显示不同	machine_name% su Password:
<i>AaBbCc123</i>	要使用实名或值替换的命令行占位符	删除文件的命令为 <code>rm filename</code> 。
<i>AaBbCc123</i>	保留未译的新词或术语以及要强调的词	这些称为 <i>Class</i> 选项。 [请注意有些强调的项目在联机时以粗体显示。]
新词术语强调	新词或术语以及要强调的词	执行 修补程序分析 。 请勿保存文件。

表 P-1 印刷约定 (续)

字体或符号	含义	示例
《书名》	书名	阅读《用户指南》的第 6 章。

命令中的 shell 提示符示例

下表列出了 C shell、 Bourne shell 和 Korn shell 的缺省系统提示符和超级用户提示符。

表 P-2 Shell 提示符

Shell	提示符
C shell	machine_name%
C shell 超级用户	machine_name#
Bourne shell 和 Korn shell	\$
Bourne shell 和 Korn shell 超级用户	#

Solaris Volume Manager 入门

《Solaris Volume Manager 管理指南》介绍如何设置和维护使用 Solaris Volume Manager 管理存储的系统，以实现高可用性、灵活性和可靠性。

本章可用作查找某些 Solaris Volume Manager 任务（如设置存储容量）信息的概括性指南，其中并没有说明使用 Solaris Volume Manager 所需的所有任务，而是概述了新增的功能，提供了查找与 Solaris Volume Manager 概念相关的常见任务的说明过程的简单方法。

本章包含以下汇总信息：

- 第 24 页中的 “Solaris Volume Manager 汇总信息—新增功能”
- 第 24 页中的 “Solaris Volume Manager 汇总信息—存储容量”
- 第 25 页中的 “Solaris Volume Manager 汇总信息—可用性”
- 第 25 页中的 “Solaris Volume Manager 汇总信息—I/O 性能”
- 第 26 页中的 “Solaris Volume Manager 汇总信息—管理”
- 第 27 页中的 “Solaris Volume Manager 汇总信息—故障排除”



注意 – 如果没有正确地使用 Solaris Volume Manager，则会销毁数据。Solaris Volume Manager 提供了一种可靠管理磁盘和磁盘上的数据的有效方法。但是，应始终维护数据备份，特别是在修改活动的 Solaris Volume Manager 配置之前。

Solaris Volume Manager 汇总信息 — 新增功能

任务	说明	参考
管理有一个或多个超过 1 TB 的组件的存储区	使用大小超过 1 TB 的物理逻辑单元号 (logical unit number, LUN)，或创建超过 1 TB 的逻辑卷。	第 45 页中的 “Solaris Volume Manager 中的多 TB 支持的概述”
将磁盘集从一个系统导入另一个系统	使用 <code>metaimport</code> 命令导入磁盘集（即使该磁盘集是在不同系统上创建的）。此命令使用扩展的设备 ID 支持，自动跟踪磁盘在已命名的磁盘集中的移动。	第 199 页中的 “导入磁盘集” 第 205 页中的 “磁盘集中的异步共享存储”
创建和管理多属主磁盘集	使用 <code>metaset -M</code> 管理 Sun Cluster 环境中的多属主磁盘集。	第 53 页中的 “与多属主磁盘集关联的任务”

Solaris Volume Manager 汇总信息 — 存储容量

任务	说明	参考
设置存储区	通过创建 RAID-0 卷或 RAID-5 卷来创建跨多片的存储区。然后，可以将 RAID-0 卷或 RAID-5 卷用于文件系统或访问原始设备的任何应用程序（如数据库）。	第 84 页中的 “如何创建 RAID-0（条带化）卷” 第 85 页中的 “如何创建 RAID-0（串联）卷” 第 104 页中的 “如何从未使用的片创建 RAID-1 卷” 第 107 页中的 “如何从文件系统创建 RAID-1 卷” 第 170 页中的 “如何创建 RAID-5 卷”
扩展现有文件系统	通过创建 RAID-0（串联）卷，然后向该卷中添加其他片，增加现有文件系统的容量。	第 87 页中的 “如何扩展现有数据的存储容量”
扩展现有的 RAID-0（串联或条带化）卷	通过将其他片串联到现有的 RAID-0 卷来扩展该卷。	第 88 页中的 “如何扩展现有的 RAID-0 卷”
扩展 RAID-5 卷	通过将其他片串联到 RAID-5 卷来扩展该卷的容量。	第 172 页中的 “如何扩展 RAID-5 卷”

任务	说明	参考
增加扩展卷上的 UFS 文件系统的大小	在 UFS 挂载时使用 <code>growfs</code> 命令扩大 UFS 的大小（同时不中断对数据的访问），以扩展文件系统。	第 241 页中的 “如何扩展文件系统”
将片或逻辑卷划分为较小的分区，打破 8 片分区的硬性限制	使用软分区划分逻辑卷或片。	第 155 页中的 “如何创建软分区”
创建文件系统	在 RAID-0（条带化或串联）卷、RAID-1（镜像）卷和 RAID-5 卷上，或者在软分区上创建文件系统。	《系统管理指南：设备和文件系统》中的第 18 章“创建 UFS、TMPFS 和 LOFS 文件系统（任务）”

Solaris Volume Manager 汇总信息 — 可用性

任务	说明	参考
最大程度地提高数据可用性	使用 Solaris Volume Manager 的镜像功能维护数据的多个副本。可以使用为数据准备的未使用的片来创建 RAID-1 卷；也可以镜像现有文件系统，包括根 (/) 和 /usr。	第 104 页中的 “如何从未使用的片创建 RAID-1 卷” 第 107 页中的 “如何从文件系统创建 RAID-1 卷”
以最低硬件成本增加数据可用性	使用 Solaris Volume Manager 的 RAID-5 卷以最低硬件成本增加数据可用性。	第 170 页中的 “如何创建 RAID-5 卷”
增加现有 RAID-1 卷或 RAID-5 卷的数据可用性	通过创建热备用池，然后将其与 RAID-1 卷或 RAID-5 卷的子镜像关联起来，来增加 RAID-1 卷或 RAID-5 卷的数据可用性。	第 184 页中的 “创建热备用池” 第 186 页中的 “将热备用池与卷相关联”

Solaris Volume Manager 汇总信息 — I/O 性能

任务	说明	参考
调整 RAID-1 卷读取和写入策略	为 RAID-1 卷指定读取和写入策略，以提高给定配置的 I/O 性能。	第 99 页中的 “RAID-1 卷读取和写入策略” 第 136 页中的 “如何更改 RAID-1 卷选项”

任务	说明	参考
优化设备性能	创建 RAID-0（条带化）卷以优化组成条带化卷的设备的 I/O 性能。可以针对随机访问或顺序访问优化交错值。	第 83 页中的 “创建 RAID-0（条带化）卷”
维护 RAID-0（条带化）中的设备性能	通过将新组件串联到空间不足的条带化卷或串联卷来扩展该卷。条带串联比片串联更有益于 I/O 性能的提高。	第 86 页中的 “扩展存储容量”

Solaris Volume Manager 汇总信息—管理

任务	说明	参考
以图形化的方式管理卷管理配置	使用 Solaris Management Console 图形用户界面 (graphical user interface, GUI) 来管理卷管理配置。	Solaris Management Console 应用程序的 Solaris Volume Manager（增强的存储）节点中的联机帮助
以图形化的方式管理片和文件系统	使用 Solaris Management Console GUI 管理磁盘和文件系统，执行对磁盘进行分区和构造 UFS 文件系统等任务。	Solaris Management Console 应用程序中的联机帮助
优化 Solaris Volume Manager	Solaris Volume Manager 的性能取决于设计完善的配置。创建后，此配置仍需要监视和调整。	第 44 页中的 “Solaris Volume Manager 配置原则” 第 238 页中的 “使用配置文件”
规划未来的扩展	由于文件系统往往会空间不足，因此可通过串联文件系统来规划未来的扩充。	第 85 页中的 “创建 RAID-0（串联）卷” 第 86 页中的 “扩展存储容量”

Solaris Volume Manager 汇总信息 — 故障排除

任务	说明	参考
替换故障片	如果磁盘出现故障，则必须替换 Solaris Volume Manager 配置中使用的片。如果 RAID-0 卷出现故障，必须使用新的片，删除并重新创建卷，然后从备份中恢复数据。可以替换和重新同步 RAID-1 和 RAID-5 卷中的片，而不会丢失数据。	第 138 页中的 “响应 RAID-1 卷组件的故障” 第 174 页中的 “如何在 RAID-5 卷中替换组件”
从引导问题中恢复	引导系统时，由于硬件问题或操作员错误，可能会出现特殊的问题。	第 300 页中的 “如何从错误的 /etc/vfstab 项中恢复” 第 310 页中的 “如何从状态数据库副本不足中恢复” 第 303 页中的 “如何从引导设备故障中恢复”

存储管理概念

本章简要介绍了一些常见的存储管理概念。

本章包含以下信息：

- 第 29 页中的 “存储管理介绍”
- 第 30 页中的 “配置规划指南”
- 第 32 页中的 “常规性能指南”
- 第 33 页中的 “随机 I/O 和顺序 I/O 优化”

存储管理介绍

选择管理存储的方式决定了控制系统上用于存储活动数据的设备的方式。要使活动的数据有用，则这些数据必须可用，且始终保持不变，即使是在出现意外事件（如硬件或软件故障）后也是如此。

存储硬件

有许多可用于存储数据的不同设备。选择最能满足存储需求的设备主要取决于三个因素：

- 性能
- 可用性
- 成本

使用 Solaris Volume Manager 有助于管理性能、可用性和成本三者的权衡因素。通常，使用 Solaris Volume Manager 可以减少许多权衡因素。

Solaris Volume Manager 可以与运行 Solaris 操作系统的任何系统上的任何受支持存储设备很好地协作。

RAID 级别

RAID 是廉价/独立磁盘冗余阵列 (Redundant Array of Inexpensive/Independent Disks) 的首字母缩略词。RAID 是指对于用户显示为单个大型磁盘驱动器的一组磁盘，称为阵列或卷。根据配置，此阵列提供增强的可靠性、响应时间或存储容量。

从技术上讲，有六个 RAID 级别 (0-5)。每个级别代表一种在确保数据冗余性的同时分布数据的方法。（虽然 RAID 级别 0 不提供数据冗余性，但通常也作为一种 RAID 类别。RAID 级别 0 为使用的大多数 RAID 配置提供基础。）极少有存储环境支持 RAID 级别 2、3 和 4，因此，本书没有介绍这些级别。

Solaris Volume Manager 支持以下 RAID 级别：

- **RAID 级别 0**—尽管条带和串联都不提供冗余性，但这些卷通常称为 RAID-0。实质上，数据分布于相对较小且大小相等的分段中，这些分段交替且均匀地分配于多个物理磁盘中。任何单个驱动器故障都会导致数据丢失。RAID-0 提供了较高的数据传输率和较高的 I/O 吞吐量，但与单个磁盘相比，可靠性和可用性都比较低。
- **RAID 级别 1**—镜像过程使用等量磁盘容量存储数据和一个数据副本（镜像）。数据在两个或更多物理磁盘上进行复制（即镜像）。可以同时从两个驱动器中读取数据，这表示每个驱动器都能为任何请求提供服务，从而提供改进的性能。如果一个物理磁盘出现故障，可以继续使用镜像，从而不会造成性能降低或数据丢失。

Solaris Volume Manager 支持 RAID-0+1 和（透明地）RAID-1+0 镜像，这取决于基础卷。有关详细信息，请参见第 94 页中的“提供 RAID-1+0 和 RAID-0+1”。

- **RAID 级别 5**—RAID-5 使用条带化将数据分布于阵列中的磁盘上。RAID-5 还将记录奇偶校验信息，从而提供一定程度的数据冗余性。RAID-5 卷不会因基础设备的故障而发生故障。如果将 RAID-5 卷与热备件结合使用，则该卷可以经受多种故障，而本身不会出现任何故障。若在运行时有设备发生故障，RAID-5 卷的性能会明显降低。

在 RAID-5 模型中，每个设备都有一个包含奇偶校验条带的区域和其他包含数据的区域。奇偶校验分布于阵列中的所有磁盘上，可以缩短写入时间。由于不必等到专门分配的奇偶校验磁盘接受数据后再进行写入，因而缩短了写入时间。

配置规划指南

规划存储管理配置时，请牢记，对于任何给定配置，都需要在性能、可用性和硬件成本之间进行权衡。可能需要试用不同类型的存储器，以便确定最适合您的配置的存储设备。

本节提供了如何选择使用以下类型卷的指南：

- RAID-0（串联和条带化）卷
- RAID-1（镜像）卷
- RAID-5 卷

- 软分区
- 在 Solaris Volume Manager 卷上构造的文件系统

选择存储设备

实现存储管理方法之前，需要确定要使用的存储设备类型。这组指南对各种类型的存储设备进行了比较，可帮助您做出选择。其他组的指南适用于在 Solaris Volume Manager 中实现的特定类型的存储设备。有关详细信息，请参见有关每种卷类型的特定章节。

注- 此处列出的存储类型相互之间并不排斥。可以组合使用这些卷来实现多个目标。例如，可以首先创建 RAID-1 卷来实现冗余性。接下来，可以在 RAID-1 卷上创建软分区来增加独立文件系统的可能数目。

下表提供了各种类型的存储类型可用功能的比较。

表 2-1 存储类型的比较

要求	RAID-0（串联）	RAID-0（条带）	RAID-1（镜像）	RAID-5	软分区
是否提供冗余数据	否	否	是	是	否
是否增强了读取性能	否	是	取决于基础设备	是	否
是否增强了写入性能	否	是	否	否	否
每个设备是否超过 8 个片	否	否	否	否	是
是否有更大的可用存储空间	是	是	否	是	否

下表概括了 RAID-1 和 RAID-5 卷在写入操作、随机读取和硬件成本三者之间的权衡。

表 2-2 优化冗余存储

	RAID-1（镜像）	RAID-5
写入操作	速度较快	速度较慢
随机读取	速度较快	速度较慢

表 2-2 优化冗余存储 (续)

	RAID-1 (镜像)	RAID-5
硬件成本	较高	较低

以下对表中的信息进行了概括：

- RAID-0 卷（条带化和串联）和软分区不提供任何数据冗余性。
- 串联非常适用于小型随机 I/O 操作。
- 条带化非常适合于大型顺序 I/O 操作和随机 I/O 操作。
- 镜像可能会提高读取性能，但在镜像中写入性能始终会降低。
- 由于 RAID-5 卷具有读取-修改-写入特性，因此写入量超过 20% 的卷不应该为 RAID-5。如果需要冗余性，请考虑使用镜像。
- RAID-5 写入不如镜像写入的速度快，而镜像写入不如未保护写入的速度快。
- 软分区对于管理非常大的存储设备非常有用。

注 - 除这些常用存储选项之外，有关使用 Solaris Volume Manager 支持冗余设备的更多信息，请参见第 43 页中的“热备用池”。

常规性能指南

设计存储配置时，请考虑以下性能指南：

- 一般情况下，条带化的性能最佳，但条带化不提供数据冗余性。对于进行大量写入操作的应用程序，一般情况下，RAID-1 卷比 RAID-5 卷的性能更好。
- RAID-1 和 RAID-5 卷都可以增加数据可用性，但一般情况下，这两种卷的写入操作性能都比较低。镜像可以提高随机读取性能。
- RAID-5 卷比 RAID-1 卷的硬件成本更低，而 RAID-0 卷没有其他硬件成本。
- 条带和 RAID-5 卷都将数据分布于多个磁盘驱动器中并且有助于平衡 I/O 负载。
- 确定最常访问的数据，并使用镜像或条带化来增加数据的访问带宽。
- 使用可用的性能监视功能和通用工具（如 iostat 命令）以确定最常访问的数据。确定最常访问的数据之后，可以使用条带化、RAID-1 卷或 RAID-5 卷来增加此数据的访问带宽。
- 多次更改软分区大小，可能会降低软分区的性能。
- 在写操作方面，RAID-5 卷比条带的性能更低。此性能降低是由于计算和存储 RAID-5 卷奇偶校验所需的多项 I/O 操作导致的。
- 对于原始随机 I/O 读取，条带和 RAID-5 卷相当。条带和 RAID-5 卷都将数据分布在多个磁盘中。对于 RAID-5 卷，读取过程中并不进行奇偶校验计算（出现片故障后除外）。

- 对于原始随机 I/O 写入，条带优于 RAID-5 卷。

有关特定于 Solaris Volume Manager 的配置原则，请参见第 44 页中的“[Solaris Volume Manager 配置原则](#)”。

随机 I/O 和顺序 I/O 优化

本节说明用于优化配置的策略。

如果不知道在将要创建的 Solaris Volume Manager 卷上是顺序 I/O 还是随机 I/O 占支配地位，请勿实现这些性能调整提示。如果错误地实现这些提示，则会降低性能。

以下优化建议假设您要优化 RAID-0 卷。通常需要优化 RAID-0 卷，然后镜像该卷以提供最佳性能和数据冗余性。

随机 I/O

在随机 I/O 环境中（如用于数据库和通用文件服务器的环境），所有的磁盘都应花费相同的时间为 I/O 请求提供服务。

例如，假设有 40 GB 存储用于数据库应用程序。如果跨四个 10 GB 的磁盘轴进行条带化，并且 I/O 是随机且均匀地分散于卷中，则每个磁盘的繁忙程度一样，一般情况下会提高性能。

磁盘上最大随机 I/O 性能的目标是 35% 或更低的使用率（由 `iostat` 命令报告）。通常，磁盘使用率超过 65% 就会出现严重问题。磁盘使用率超过 90% 就会出现严重问题。解决磁盘使用率值太高的办法就是创建包含更多磁盘（轴）的新 RAID-0 卷。

注 - 仅将其他磁盘连接到现有卷不能提高性能。可以使用理想的参数创建新卷，从而优化性能。

条带的交错大小并不重要，因为您只想将数据分布于所有磁盘中。任何大于一般 I/O 请求的交错值便已足够。

顺序访问 I/O

可以在顺序 I/O 环境（如整表扫描占支配地位的 DBMS 服务器，以及数据非常密集的环境中的 NFS 服务器）中优化配置的性能。要充分利用顺序 I/O 环境，请相对于一般 I/O 请求的大小将交错值设置为较小的值。

例如，假设一般 I/O 请求大小为 256 KB，且跨 4 个轴进行条带化。本示例中的条带单元大小的最佳选择将为：

$256 \text{ KB} / 4 = 64 \text{ KB}$ ，或更小

此策略确保了一般 I/O 请求跨多个磁盘轴分布，从而增加了连续带宽。

注 - 在顺序 I/O 环境中，查找时间和旋转时间实际上都为零。优化顺序 I/O 时，磁盘的内部传输率最为重要。

在顺序的应用程序中，一般 I/O 大小通常很大（大于 128 KB 或者甚至大于 1 MB）。假设应用程序的一般 I/O 请求大小为 256 KB，并假设跨 4 个磁盘轴进行条带化，因而：

$256 \text{ KB} / 4 = 64 \text{ KB}$

因此，交错值的最佳选择将为 32–64 KB。

Solaris Volume Manager 概述

本章介绍 Solaris Volume Manager 的总体结构。本章包含以下信息：

- 第 35 页中的 “Solaris Volume Manager 中的新增功能”
- 第 35 页中的 “Solaris Volume Manager 介绍”
- 第 38 页中的 “Solaris Volume Manager 要求”
- 第 38 页中的 “Solaris Volume Manager 组件概述”
- 第 44 页中的 “Solaris Volume Manager 配置原则”
- 第 44 页中的 “有关创建 Solaris Volume Manager 组件的概述”
- 第 45 页中的 “Solaris Volume Manager 中的多 TB 支持的概述”
- 第 46 页中的 “升级到 Solaris Volume Manager”

Solaris Volume Manager 中的新增功能

本节介绍了此 Solaris 发行版中在使用 Solaris Volume Manager 方面的新增功能。

有关 Solaris 新增功能的完整列表以及 Solaris 发行版的描述，请参见《Solaris 10 新增功能》。

Solaris Volume Manager 介绍

Solaris Volume Manager 是用于管理大量磁盘及磁盘上数据的软件产品。尽管 Solaris Volume Manager 产品的用途有很多，但主要任务包括：

- 增加存储容量
- 提高数据可用性
- 简化大型存储设备的管理

在某些情况下，Solaris Volume Manager 还可以提高 I/O 性能。

有关 Solaris 操作系统所支持的磁盘类型的信息，请参见《系统管理指南：设备和文件系统》中的第 11 章“管理磁盘（概述）”。

Solaris Volume Manager 如何管理存储

Solaris Volume Manager 使用虚拟磁盘来管理物理磁盘及其关联的数据。在 Solaris Volume Manager 中，虚拟磁盘称为**卷**。出于历史原因，某些命令行实用程序也将卷称为**元设备**。

在应用程序或文件系统看来，从功能方面讲，卷与物理磁盘完全相同。Solaris Volume Manager 可将定向到卷的 I/O 请求转换成定向到基础成员磁盘的 I/O 请求。

Solaris Volume Manager 卷是由磁盘片或其他 Solaris Volume Manager 卷构建的。创建卷的简单方法是使用内置于 Solaris Management Console 中的图形用户界面 (graphical user interface, GUI)。Solaris Management Console 中增强的存储工具为您展示了所有现有卷的视图。按照向导中的步骤，您可以方便地生成任何种类的 Solaris Volume Manager 卷或组件。您还可以使用 Solaris Volume Manager 命令行实用程序生成和修改卷。

例如，如果需要更多存储容量作为单个卷，可以使用 Solaris Volume Manager 使系统将片集合视为一个较大的卷。使用这些片创建卷之后，可以立即开始使用该卷，就像使用任何“真正”的片或设备一样。

有关卷的更详细的介绍，请参见第 39 页中的“卷概述”。

Solaris Volume Manager 通过使用 RAID-1（镜像）卷和 RAID-5 卷可以提高数据的可靠性和可用性。Solaris Volume Manager 热备件可以进一步提高镜像卷和 RAID-5 卷的数据可用性。

设置配置后，可以使用 Solaris Management Console 中增强的存储工具来报告其操作。

如何管理 Solaris Volume Manager

使用以下方法之一来管理 Solaris Volume Manager：

- Solaris Management Console — 此工具提供用于管理卷管理功能的 GUI。可使用 Solaris Management Console 中增强的存储工具。有关增强的存储工具的示例，请参见图 3-1。此界面提供 Solaris Volume Manager 组件（包括卷、热备用池和状态数据库副本）的图形视图。此界面提供对 Solaris Volume Manager 组件的基于向导的处理功能，使您可以快速配置磁盘或更改现有配置。
- 命令行 — 可以使用多个命令来执行卷管理功能。Solaris Volume Manager 核心命令以 `meta` 开头，例如 `metainit` 和 `metastat` 命令。有关 Solaris Volume Manager 命令的列表，请参见附录 B。

注 – 请不要尝试同时使用命令行和 GUI 来管理 Solaris Volume Manager。对配置进行的更改可能会发生冲突，致使其行为不可预测。可以使用这两种工具来管理 Solaris Volume Manager，但不能同时使用。

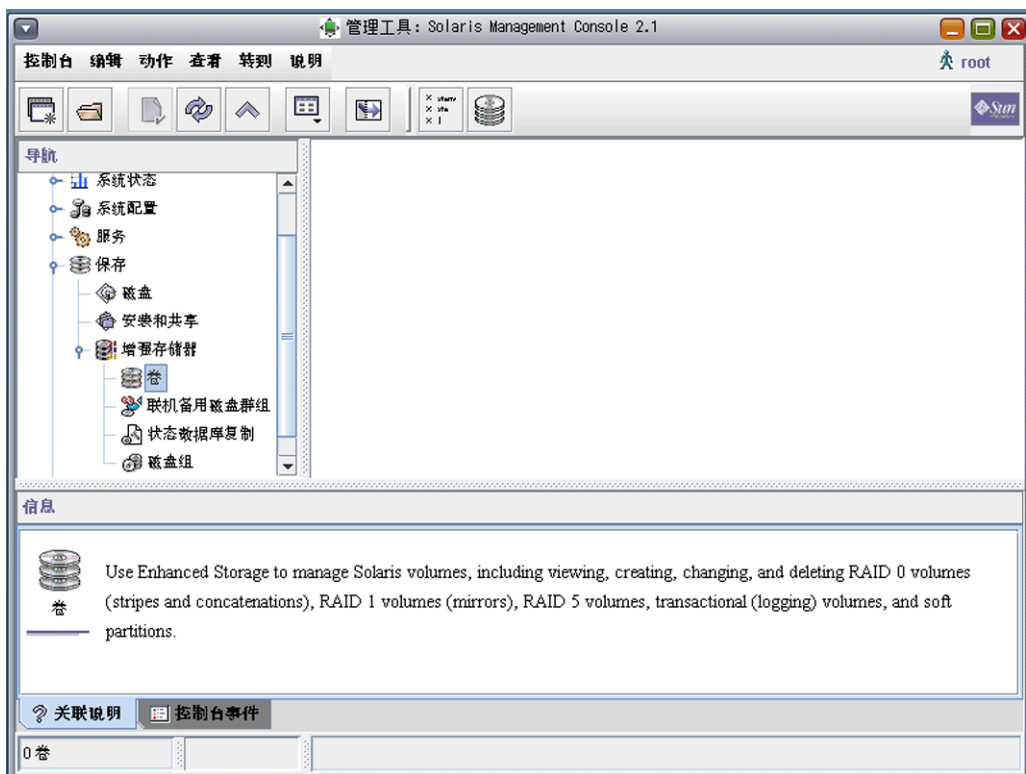


图 3-1 Solaris Management Console 中增强的存储工具 (Solaris Volume Manager) 视图

▼ 如何访问 Solaris Volume Manager 图形用户界面 (Graphical User Interface, GUI)

Solaris Volume Manager GUI（增强的存储）是 Solaris Management Console 的一部分。要访问 GUI，请按照以下说明操作：

- 1 使用以下命令在主机系统上启动 Solaris Management Console ：
`% /usr/sbin/smc`
- 2 在“导航”窗格中双击“本计算机”。
- 3 在“导航”窗格中双击“保存”。
- 4 在“导航”窗格中双击“增强存储”，以装入 Solaris Volume Manager 工具。

- 5 如果系统提示您登录，请以超级用户或具备等效访问权限的用户身份登录。
- 6 双击用于管理卷、热备用池、状态数据库副本和磁盘集的相应图标。

提示 – Solaris Management Console 中的所有工具都在控制台窗口底部或向导面板左侧显示信息。在任何时候选择“帮助”，都可查找有关在此界面中执行任务的其他信息。

Solaris Volume Manager 要求

Solaris Volume Manager 要求包括以下内容：

- 必须具备超级用户权限才能管理 Solaris Volume Manager。通过 Solaris Management Console 中的“用户配置文件”功能授予的等效权限允许通过 Solaris Management Console 进行管理。但是，只有超级用户才能使用 Solaris Volume Manager 命令行界面。
- 使用 Solaris Volume Manager 创建卷之前，Solaris Volume Manager 系统上必须存在状态数据库副本。状态数据库副本包含所有卷、热备件和磁盘集的配置和状态信息。应该至少存在三个副本；而且，为了获得最大可靠性，这些副本应置于不同的控制器和不同的磁盘上。有关状态数据库副本的更多信息，请参见第 63 页中的“关于 Solaris Volume Manager 状态数据库和副本”。有关如何创建状态数据库副本的说明，请参见第 70 页中的“创建状态数据库副本”。

Solaris Volume Manager 组件概述

使用 Solaris Volume Manager 创建的五种基本类型的组件为卷、软分区、磁盘集、状态数据库副本和热备用池。下表概述了这些 Solaris Volume Manager 功能。

表 3–1 Solaris Volume Manager 功能汇总

Solaris Volume Manager 功能	定义	目的	更多信息
<ul style="list-style-type: none">■ RAID-0 卷（条带化卷、串联卷、串联的条带化卷）■ RAID-1（镜像）卷■ RAID-5 卷	一组物理片，在系统中显示为单个逻辑设备	用于增加存储容量、改善性能或提高数据可用性。	第 39 页中的“卷概述”
软分区	物理片或逻辑卷的细分，提供更小，更易于管理的存储单元	用于提高大型存储卷的可管理性。	第 12 章

表 3-1 Solaris Volume Manager 功能汇总 (续)

Solaris Volume Manager 功能	定义	目的	更多信息
状态数据库 (状态数据库副本)	包含所有卷、热备件和磁盘集的配置和状态信息的数据库。创建状态数据库副本之前，Solaris Volume Manager 无法操作。	用于存储有关 Solaris Volume Manager 配置状态的信息	第 42 页中的 “状态数据库和状态数据库副本”
热备用池	保留的片 (热备件) 集合。子镜像或 RAID-5 卷组件出现故障时，系统会使用这些片进行替换。	用于提高 RAID-1 和 RAID-5 卷的数据可用性。	第 43 页中的 “热备用池”
磁盘集	单独名称空间中的一组共享的磁盘驱动器，其中包含可由多台主机不同时共享的卷和热备件	用于提供数据冗余性和数据可靠性，并提供单独的名称空间以便更易于管理。	第 43 页中的 “磁盘集”

卷概述

卷是一组物理片，在系统中显示为单个逻辑设备。在标准的 UNIX® 术语中，卷实际上是伪（或虚拟）设备。

注 - 以前，Solstice DiskSuite™ 产品将这些逻辑设备称为元设备。但是，为了简单化和标准化，本书将这些设备称为卷。

卷的类别

可以将卷作为 RAID-0（串联或条带化）卷、RAID-1（镜像）卷、RAID-5 卷或软分区进行创建。

可以使用 Solaris Management Console 中增强的存储工具或命令行实用程序来创建和管理卷。

下表汇总了各种类别的卷。

表 3-2 卷的类别

卷	说明
RAID-0（条带化卷或串联卷）	可以直接使用，也可用作镜像的基本生成块。RAID-0 卷不直接提供数据冗余性。
RAID-1（镜像）	通过维护多个副本来复制数据。RAID-1 卷由一个或多个称为子镜像的 RAID-0 卷组成。

表 3-2 卷的类别 (续)

卷	说明
RAID-5	通过使用奇偶校验信息来复制数据。在磁盘出现故障时，可以使用可用数据和奇偶校验信息重新生成缺少的数据。一般情况下，RAID-5 卷是由片组成的。将大小相当于一片的空间用来存储奇偶校验信息，但奇偶校验分布于 RAID-5 卷的所有片上。
软分区	将片或逻辑卷分成一个或多个更小的可扩展卷。

如何使用卷

可以使用卷来提高存储容量、性能和数据可用性。在某些情况下，卷还可以提高 I/O 性能。在功能方面，卷与片特性相同。卷与片很相像，所以对于最终用户、应用程序和文件系统，卷是透明的。就像访问物理设备一样，可以通过块或原始设备名称来访问卷。根据使用的是块还是原始设备，卷名称也会有所不同。有关卷名称的详细信息，请参见第 42 页中的“卷名称”。

可以对卷使用大多数文件系统命令，包括 `mkfs`、`mount`、`umount`、`ufsdump`、`ufsrestore` 等。但是，不能使用 `format` 命令。只要卷包含挂载的文件系统，就可以对该卷进行读取、写入和复制文件等操作。

示例—包含两个片的卷

图 3-2 显示一个包含两个片的卷，一个片来自磁盘 A，另一个片来自磁盘 B。应用程序或 UFS 会像处理一个物理磁盘那样处理卷。通过向卷中添加更多片，可以增加其存储容量。

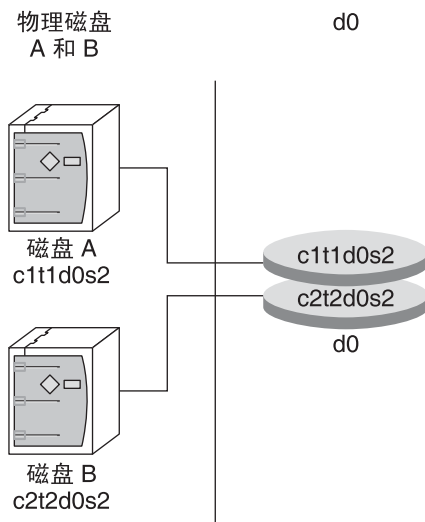


图 3-2 卷、物理磁盘和片之间的关系

使用 growfs 命令扩展卷和磁盘空间

使用 Solaris Volume Manager，可以通过添加其他片来扩展卷。可以使用 Solaris Management Console 中增强的存储工具 或命令行界面向现有卷添加片。

在不停止或备份系统的情况下，可以扩展卷中包含的已挂载或未挂载的 UFS 文件系统。不过，备份数据总是有益无害的。扩展卷后，请使用 `growfs` 命令来扩充文件系统。

注 - 扩展文件系统后，就不能缩小文件系统的大小。无法缩小文件系统的大小是 UFS 的限制。同样，增大 Solaris Volume Manager 分区后，就不能再将其缩小。

使用原始卷的应用程序和数据库必须自备“增加”扩展空间的方法，以便应用程序可以识别它。Solaris Volume Manager 不提供此功能。

可以采用以下方式扩展卷中的磁盘空间：

- 向 RAID-0 卷添加一个或多个片
- 向 RAID-1 卷的所有子镜像添加一个或多个片
- 向 RAID-5 卷添加一个或多个片
- 通过添加基础组件中的空间扩展软分区

`growfs` 命令可以扩展 UFS 文件系统，而不会中断服务或丢失数据。但是，在 `growfs` 命令运行的同时会暂停对卷的写访问。可以将文件系统的大小扩展到包含该文件系统的片或卷的大小。

通过对 `growfs` 命令使用 `-s size` 选项，可以扩展文件系统以便仅使用所增加的磁盘空间的部分。

注 - 扩展镜像时，先向镜像的基础子镜像添加空间。然后，再对 RAID-1 卷运行 `growfs` 命令。一般规则是向基础设备添加空间，并对顶层设备运行 `growfs` 命令。

卷名称

就像物理片一样，卷的逻辑名称也会出现在文件系统中。逻辑卷名在块设备的 `/dev/md/dsk` 目录和原始设备的 `/dev/md/rdsk` 目录中具有对应的项。在任何 `meta*` 命令中，通常使用缩写形式的卷名（如 `d1`），而不用指定完整的卷名（如 `/dev/md/dsk/volume-name`）。一般情况下，只要卷当前没有被使用，且新名称没有被其他卷使用，就可以对卷进行重命名。有关更多信息，请参见第 236 页中的“交换卷名称”。

最初，卷名称必须以字母“d”开头，后跟一个数字（例如 `d0`）。此格式现在仍可使用。下面是使用“d*”命名构造的卷名称示例：

```
/dev/md/dsk/d0      块卷 d0
/dev/md/dsk/d1      块卷 d1
/dev/md/rdsk/d126   原始卷 d126
/dev/md/rdsk/d127   原始卷 d127
```

卷名称命名原则

命名卷名称时使用一个统一的标准可以简化管理并使您可以快速识别卷类型。以下是几条建议：

- 为每种类型的卷指定范围。例如，为 RAID-1 卷指定数字 0–20，为 RAID-0 卷指定数字 21–40，等等。
- 通过命名确定镜像之间的关系。例如，使用以零 (0) 结尾的数字命名镜像，使用以一 (1)、二 (2) 等结尾的数字命名子镜像。例如，可以按照以下方式命名镜像：镜像 `d10`、子镜像 `d11` 和 `d12`；镜像 `d20`、子镜像 `d21`、`d22`、`d23` 和 `d24`。
- 使用将片编号和磁盘编号映射到卷编号的命名方法。

状态数据库和状态数据库副本

状态数据库是用于存储有关 Solaris Volume Manager 配置状态信息的数据库。状态数据库记录并跟踪对配置所做的更改。配置或状态发生更改时，Solaris Volume Manager 会自动更新状态数据库。例如，创建新卷是一种配置更改，而子镜像失败则是一种状态更改。

实际上，状态数据库是多个复制的数据库副本的集合。每个副本（称为**状态数据库副本**）都可以确保状态数据库中的数据始终有效。多个状态数据库副本可以防止发生单点故障时丢失数据。状态数据库可以跟踪所有已知状态数据库副本的位置和状态。

在创建状态数据库及其状态数据库副本之前，Solaris Volume Manager 无法进行操作。Solaris Volume Manager 配置中必须具有运行状态数据库。

当设置配置时，您可以将状态数据库副本放在以下位置之一：

- 在专用的片上
- 在以后将成为卷的组成部分的片上

Solaris Volume Manager 可以识别包含状态数据库副本的片，如果该片位于某个卷中而且正在使用，则会自动跳过其中的副本。为状态数据库副本保留的那部分片不应用于任何其他用途。

一个片上可以存储多个状态数据库副本。但是这样做，在发生单点故障时系统可能更容易受到损坏。

Solaris 操作系统在删除所有状态数据库副本后仍然可以继续正常工作。但是，如果在磁盘上不存在现有状态数据库副本的情况下重新引导系统，系统会丢失所有的 Solaris Volume Manager 配置数据。

热备用池

热备用池是 Solaris Volume Manager 保留的用于自动替换出故障组件的片（**热备件**）的集合。可以在子镜像或 RAID-5 卷中使用这些热备件。热备件为 RAID-1 和 RAID-5 卷提供增强的数据可用性。可以使用 Solaris Management Console 中增强的存储工具 或命令行界面来创建热备用池。

出现组件错误时，Solaris Volume Manager 会查找第一个等于或大于故障组件大小的热备件。如果找到符合条件的热备件，Solaris Volume Manager 会自动替换组件并重新同步数据。如果未在热备件列表中找到适当大小的片，则认为子镜像或 RAID-5 卷已失败。有关更多信息，请参见第 16 章。

磁盘集

磁盘集是一组物理存储卷，其中包含逻辑卷和热备件。卷和热备用池必须在该磁盘集中的驱动器上生成。在磁盘集中创建卷后，可以像使用物理片一样使用该卷。

磁盘集在群集环境中提供数据可用性。如果某台主机出现故障，其他主机可以接管出故障主机的磁盘集。（这种类型的配置称为**故障转移配置**。）此外，可以使用磁盘集来帮助管理 Solaris Volume Manager 名称空间，并实现对网络连接存储设备的随时访问。

有关更多信息，请参见第 18 章。

Solaris Volume Manager 配置原则

设计不完善的 Solaris Volume Manager 配置可能会降低性能。本节提供有关实现 Solaris Volume Manager 的良好性能的提示。有关存储配置性能原则的信息，请参见第 32 页中的“常规性能指南”。

一般原则

- **磁盘和控制器**—将卷中的驱动器置于单独的驱动器路径上（对于 SCSI 驱动器，置于单独的主机适配器上）。将 I/O 负载分布在多个控制器中可以提高卷的性能和可用性。
- **系统文件**—绝对不要编辑或删除 `/etc/lvm/mddb.cf` 或 `/etc/lvm/md.cf` 文件。确保定期备份这些文件。
- **卷完整性**—如果将片定义为卷，请不要将基础片用于任何其他目的，包括将片用作转储设备。
- **有关磁盘和分区的信息**—保留 `prtvtoc` 和 `metastat -p` 命令输出的副本，以备需要重新格式化坏磁盘或重新创建 Solaris Volume Manager 配置时使用。

文件系统挂载原则

请不要在卷的基础片上挂载文件系统。如果该片将用于任何种类的卷，则不得将该片作为文件系统进行挂载。如果可能，请在激活卷之前先取消挂载要用作卷的任何物理设备。

有关创建 Solaris Volume Manager 组件的概述

创建 Solaris Volume Manager 组件时，可以为逻辑 Solaris Volume Manager 名称（如 `d0`）指定物理片。可以创建的 Solaris Volume Manager 组件包括以下各项：

- 状态数据库副本
- 卷（RAID-0 [条带、串联]、RAID-1 [镜像]、RAID-5 和软分区）
- 热备用池
- 磁盘集

注—有关如何命名卷的建议，请参见第 42 页中的“卷名称”。

创建 Solaris Volume Manager 组件的先决条件

以下是创建 Solaris Volume Manager 组件的先决条件：

- 创建初始状态数据库副本。如果尚未执行此操作，请参见第 70 页中的“[创建状态数据库副本](#)”。
- 找出可供 Solaris Volume Manager 使用的片。如有必要，请使用 `format` 命令、`fmthard` 命令或 Solaris Management Console 对现有磁盘重新分区。
- 确保您具有超级用户权限。
- 具有所有数据的最新备份。
- 如果您正在使用 GUI，请启动 Solaris Management Console 并导航到 Solaris Volume Manager 功能。有关信息，请参见第 37 页中的“[如何访问 Solaris Volume Manager 图形用户界面 \(Graphical User Interface, GUI\)](#)”。

Solaris Volume Manager 中的多 TB 支持的概述

从 Solaris 9 4/03 发行版开始，Solaris Volume Manager 支持在运行 64 位内核的系统上使用大于 1 TB (Tbyte) 的存储设备和逻辑卷。

注 - 请使用 `isainfo -v` 确定系统是否正在运行 64 位内核。如果出现字符串 "64-bit"，则说明您正在运行 64 位内核。

使用 Solaris Volume Manager，可以执行以下操作：

- 创建、修改和删除基于（或使用）大小超过 1 TB 的逻辑存储单元 (logical storage unit, LUN) 生成的逻辑卷。
- 创建、修改和删除大小超过 1 TB 的逻辑卷。

系统自动支持大的卷。如果创建大于 1 TB 的设备，则 Solaris Volume Manager 会对其进行相应配置，而无需用户干预。

大卷支持限制

Solaris Volume Manager 仅支持在 Solaris 9 4/03 或更高发行版的系统上运行 64 位内核时使用大卷（大于 1 TB）。在以前的 Solaris 9 发行版上的 32 位内核下运行包含大卷的系统将影响 Solaris Volume Manager 的功能。具体来说，请注意以下几点：

- 如果在 32 位 Solaris 9 4/03 或更高版本内核下重新引导包含大卷的系统，则可以通过 `metastat` 输出查看大卷，但不能访问、修改或删除这些卷。此外，不能创建新的大卷。大卷上的任何卷或文件系统也都不可用。

- 如果在 Solaris 9 4/03 之前的 Solaris 发行版下重新引导包含大卷的系统，则 Solaris Volume Manager 不会启动。在其他版本的 Solaris 平台上运行 Solaris Volume Manager 之前，必须先删除所有的大卷。



注意 – 如果期望使用 32 位内核运行 Solaris 软件，或者希望使用 Solaris 9 4/03 发行版之前的 Solaris OS 版本，请不要创建大卷。

使用大卷

所有的 Solaris Volume Manager 命令都适用于大卷。不需要做任何语法变更或执行任何特殊任务，即可利用大卷支持。因此，熟悉 Solaris Volume Manager 的系统管理员可以直接使用 Solaris Volume Manager 大卷。

提示 – 如果创建大卷，随后确定需要在以前的 Solaris 发行版下使用 Solaris Volume Manager，或需要在 32 位 Solaris 9 4/03 或更高版本内核下运行，则需要删除大卷。请在 64 位内核下使用 `metaclear` 命令从 Solaris Volume Manager 配置中删除大卷，然后在以前的 Solaris 发行版或 32 位内核下重新引导系统。

升级到 Solaris Volume Manager

Solaris Volume Manager 完全支持从 Solstice DiskSuite 版本 4.1、4.2 和 4.2.1 进行无缝升级。请确保所有的卷都处于 Okay（正常）状态（而不是“Needs Maintenance（需要维护）”或“Last Erred（最近出错）”状态），而且任何热备件都没有被使用。您不需要对 Solaris Volume Manager 执行任何其他特殊任务，即可使升级顺利进行，也就是说，不必更改配置或细分引导镜像。升级系统时，将显示 Solstice DiskSuite 配置，并且升级后通过 Solaris Volume Manager 工具可以访问该配置。

Solaris 10 OS 引入了服务管理工具 (Service Management Facility, SMF)，该工具提供了用于扩充传统 UNIX 启动脚本、`init` 运行级和配置文件的基础结构。从以前的 Solaris OS 版本升级时，请验证与 Solaris Volume Manager 关联的 SMF 服务是否处于联机状态。如果 SMF 服务没有联机，管理 Solaris Volume Manager 时可能会遇到问题。

要检查与 Solaris Volume Manager 关联的 SMF 服务，请使用以下形式的 `svcs` 命令：

```
# svcs -a |egrep "md|meta"

disabled      12:05:45 svc:/network/rpc/mdcomm:default
disabled      12:05:45 svc:/network/rpc/metamed:default
```

```
disabled      12:05:45 svc:/network/rpc/metamh:default
online        12:05:39 svc:/system/metainit:default
online        12:05:46 svc:/network/rpc/meta:default
online        12:05:48 svc:/system/fmd:default
online        12:05:51 svc:/system/mdmonitor:default
```

如果 Solaris Volume Manager 配置仅包含本地集，则这些服务应该是联机的：

```
svc:/system/metainit
svc:/network/rpc/meta
svc:/system/mdmonitor
```

如果 Solaris Volume Manager 配置包括磁盘集，则这些其他服务应该是联机的：

```
svc:/network/rpc/metamed
svc:/network/rpc/metamh
```

如果 Solaris Volume Manager 包括多节点磁盘集，则除了已提到的其他服务以外，此服务也应该是联机的：

```
svc:/network/rpc/mdcomm
```

有关 SMF 的更多信息，请参见《系统管理指南：基本管理》中的第 14 章“管理服务（概述）”。

Solaris Volume Manager for Sun Cluster (概述)

本章概述了 Solaris Volume Manager for Sun Cluster。

本章包括以下信息：

- 第 49 页中的 “Solaris Volume Manager for Sun Cluster 介绍”
- 第 51 页中的 “多属主磁盘集概念”
- 第 54 页中的 “Solaris Volume Manager for Sun Cluster 配置”
- 第 55 页中的 “多属主磁盘集中的 RAID-1（镜像）卷”

Solaris Volume Manager for Sun Cluster 介绍

从 Solaris 9 9/04 发行版开始，Solaris Volume Manager 可以在 Sun Cluster 环境中使用多属主磁盘集来管理存储区。**多属主磁盘集**允许多个节点共享磁盘集的拥有权并同时写入共享磁盘。以前，磁盘集中的所有参与主机都可以看到共享磁盘集，但是一次只能有一个主机访问该磁盘集。多属主磁盘集适用于 Sun Cluster 以及诸如 Oracle Real Application Clusters 的应用程序。

多属主磁盘集和 Solaris Volume Manager 共享磁盘集可以同时存在于同一个节点上。但是，不支持在这两种配置之间移动磁盘集。

注 – 对于多属主磁盘集，不提供对 Solaris Volume Manager for Sun Cluster 设备 ID 的支持。因此，目前不支持将多属主磁盘集从一个系统导入另一个系统。

Solaris Volume Manager for Sun Cluster 创建的组件与使用 Solaris Volume Manager 创建的组件相同，包括条带、串联、镜像、软分区和热备件。Solaris Volume Manager for Sun Cluster 不支持 RAID-5 卷和事务卷。

下图显示了典型群集配置中的软件与共享存储区之间的关联。

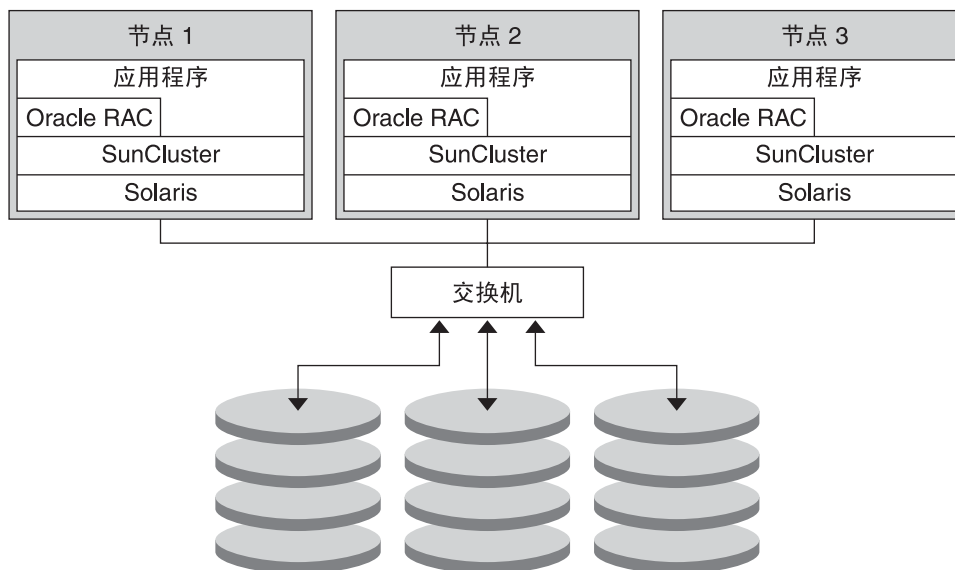


图 4-1 群集配置样例

每个节点都有本地存储区和至少一个指向共享存储区的路径。群集中的多属主磁盘集由作为 Solaris 操作系统 (Solaris Operating System, Solaris OS) 一部分的 Solaris Volume Manager for Sun Cluster 进行管理。

先决条件：多属主磁盘集功能所需的软件组件

要使用 Solaris Volume Manager for Sun Cluster，除了安装 Solaris OS 以外还必须安装以下软件：

- Sun Cluster 初始群集框架
- Sun Cluster Support for Oracle Real Application Clusters 软件
- Oracle Real Application Clusters 软件

注 - 有关设置 Sun Cluster 和 Oracle Real Application Clusters 软件的信息，请参见《Sun Cluster 软件安装指南（适用于 Solaris OS）》和《用于 Oracle Real Application Clusters 的 Sun Cluster 数据服务指南（适用于 Solaris OS）》。

多属主磁盘集概念

由 Solaris Volume Manager for Sun Cluster 管理的存储区可以划分为多属主磁盘集。多属主磁盘集允许多个节点共享磁盘集的拥有权并同时写入共享磁盘。应用程序（例如 Oracle Real Application Clusters）的实例运行在群集中的每个节点上，因此，多属主磁盘集提供了可伸缩性。由于应用程序的每个实例都可以直接访问共享存储区，因此多属主磁盘集还可以提高应用程序的性能。

注 – 多属主磁盘集功能只能在 Sun Cluster 环境中启用。**节点**是作为 Sun Cluster 系统一部分的物理计算机。

每个多属主磁盘集都与节点列表相关联。这些节点共享磁盘集的拥有权。以下 `metaset -s disk-set` 命令显示了多属主磁盘集的输出。

```
# metaset -s blue
```

Multi-owner Set name = blue, Set number = 1, Master = nodeone

Host	Owner	Member
nodeone	multi-owner	Yes
nodetwo	multi-owner	Yes

Drive Dbase

d9 Yes

d13 Yes

此输出显示了节点列表中共享磁盘集拥有权的 `nodeone` 和 `nodetwo`。此外，`nodeone` 被指定为**主节点**。

每个多属主磁盘集都有一个主节点。创建磁盘集后，向该磁盘集添加第一个磁盘的节点将成为该磁盘集的主节点。主节点可以创建、删除和更新磁盘集中的状态数据库副本。

注 – 有关状态数据库副本的更多信息，请参见第6章。

Solaris Volume Manager for Sun Cluster 支持包含不同但重叠的节点列表的磁盘集。由于每个磁盘集都有一个主节点，因此同一个群集中可以同时存在多个主节点。

metaset 命令的以下输出说明，向磁盘集添加第一个磁盘时，nodeone 将成为主节点。

```
nodeone# metaset -s red
```

```
Multi-owner Set name = red, Set number = 1, Master =
```

Host	Owner	Member
nodeone		Yes
nodetwo		Yes

```
nodeone# metaset -s red -a /dev/did/dsk/d9
```

```
nodeone# metaset -s red
```

```
Multi-owner Set name = red, Set number = 1, Master = nodeone
```

Host	Owner	Member
nodeone	multi-owner	Yes
nodetwo	multi-owner	Yes

Drive	Dbase
-------	-------

d9	Yes
----	-----

Solaris Volume Manager for Sun Cluster 支持包含不同但重叠的节点列表的磁盘集。由于每个磁盘集都有一个主节点，因此同一个群集中可以同时存在多个主节点。

与多属主磁盘集关联的任务



注意 - 配置多属主磁盘集之前，除了安装 Solaris OS 以外，还必须安装以下软件：

- Sun Cluster 初始群集框架
- Sun Cluster Support for Oracle Real Application Clusters 软件
- Oracle Real Application Clusters 软件

有关设置 Sun Cluster 和 Oracle Real Application Clusters 软件的信息，请参见《Sun Cluster 软件安装指南（适用于 Solaris OS）》和《用于 Oracle Real Application Clusters 的 Sun Cluster 数据服务指南（适用于 Solaris OS）》。

一般情况下，Solaris Volume Manager for Sun Cluster 使用同一组 Solaris Volume Manager 命令来执行与磁盘集关联的任务。已向 `metaset` 命令中添加了特定于多属主磁盘集的某些命令选项。例如，用于创建多属主磁盘集的任务需要向 `metaset` 命令中添加 `-M`。以下输出说明了如何使用 `metaset -s diskset-name -a -M -h hostname` 命令来创建多属主磁盘集。

```
# metaset -s red -a -M -h nodeone

# metaset

Multi-owner Set name = red, Set number = 1, Master =
```

Host	Owner	Member
nodeone		Yes

此外，有些 `metaset` 命令选项（如用于获取和释放磁盘集的命令）不能与多属主磁盘集一同使用。有关更多信息，请参见 `metaset(1M)` 手册页。

在 Sun Cluster 环境中使用磁盘时还存在另一个不同的任务。Sun Cluster 会为每个磁盘指定唯一的设备 ID (device ID, DID) 号。如果不想使用 `cntndn` 格式来标识磁盘，则可以使用 Sun Cluster DID 路径名 `/dev/did/dsk/dN`。变量 `N` 是由 Sun Cluster 指定的设备编号。

以下输出说明了如何使用 `metaset -s diskset-name -a disk-name` 命令向多属主磁盘集添加磁盘，以及如何使用 Sun Cluster DID 路径名标识磁盘。

```
nodeone# metaset -s red

Multi-owner Set name = red
```

Multi-owner Set name = red, Set number = 1, Master =

Host	Owner	Member
nodeone		Yes
nodetwo		Yes

nodeone# **metaset -s red -a /dev/did/dsk/d13**

nodeone# **metaset -s red**

Multi-owner Set name = red, Set number = 1, Master = nodeone

Host	Owner	Member
nodeone	multi-owner	Yes

Drive Dbase

d13 Yes

有关为 Oracle Real Application Clusters 创建多属主磁盘集的信息，请参见《用于 Oracle Real Application Clusters 的 Sun Cluster 数据服务指南（适用于 Solaris OS）》中的“在 Solaris Volume Manager for Sun Cluster 中为 Oracle Real Application Clusters 数据库创建多拥有者磁盘集”。

有关与磁盘集关联的任务，请参见[第 19 章](#)。

Solaris Volume Manager for Sun Cluster 配置

Solaris Volume Manager for Sun Cluster 支持以下配置：

- Solaris Volume Manager for Sun Cluster 最多支持 32 个磁盘集。这些磁盘集可以是多属主磁盘集、共享磁盘集和本地磁盘集的任意组合。

注 – 有关不同类型磁盘集的更多信息，请参见第 196 页中的“磁盘集类型”。

- 每个多属主磁盘集最多可支持每磁盘集 8192 个卷。
- 状态数据库副本的缺省大小为 16 MB。最小值为 16 MB。最大值为 256 MB。

Sun Cluster Support for Oracle Real Application Clusters 的许多扩展属性指定了重新配置过程中各步骤的超时。有关设置超时的详细信息，请参阅《用于 Oracle Real Application Clusters 的 Sun Cluster 数据服务指南（适用于 Solaris OS）》中的“调整 Sun Cluster Support for Oracle Real Application Clusters”。

多属主磁盘集中的 RAID-1（镜像）卷

在多属主磁盘集中创建的 RAID-1 卷（即镜像卷）与 Solaris Volume Manager 共享磁盘集中的 RAID-1 卷功能相同。但是，多属主磁盘集中的 RAID-1 卷具有一些其他功能。

多属主磁盘集的镜像拥有权

镜像拥有权的概念特定于多属主磁盘集。与 Solaris Volume Manager 共享磁盘集中的 RAID-1 卷不同，多属主磁盘集中的 RAID-1 卷通常具有与之关联的属主。镜像卷的拥有权由卷管理器进行选择。卷的属主是在磁盘集的节点列表中指定的节点之一。只有 RAID-1 卷的属主才能写入卷。如果非属主节点要写入卷，则拥有权将切换到执行写操作的节点。metastat -s diskset-name 命令的以下输出说明 RAID-1 卷 d24 的属主是 nodeone。

```
# metastat -s red

red/d24: Mirror

    Submirror 0: red/d20

        State: Okay

    Submirror 1: red/d21

        State: Okay

    Pass: 1

    Read option: roundrobin (default)

    Write option: parallel (default)
```

Resync option: optimizedresync

Owner: nodeone

Size: 825930 blocks (403 MB)

数据管理和恢复过程

与 Solaris Volume Manager 中的 RAID-1 卷一样，Solaris Volume Manager for Sun Cluster 中的 RAID-1 卷也会执行操作，从而确保数据一致性。Solaris Volume Manager for Sun Cluster 为 RAID-1 卷提供了两个数据管理和恢复选项。

Solaris Volume Manager for Sun Cluster 的优化的重新同步

Solaris Volume Manager for Sun Cluster 中优化的重新同步与 Solaris Volume Manager 中优化的重新同步功能相同。但是，在多属主磁盘集中，将重新同步选项设置为优化的重新同步的 RAID-1 卷始终具有镜像属主。`metastat -s diskset-name` 命令的以下输出给出了设置为 `optimizedresync`（优化的重新同步）的重新同步选项。

```
# metastat -s red
```

```
red/d24: Mirror
```

```
Submirror 0: red/d20
```

```
State: Okay
```

```
Submirror 1: red/d21
```

```
State: Okay
```

```
Pass: 1
```

```
Read option: roundrobin (default)
```

```
Write option: parallel (default)
```

```
Resync option: optimizedresync
```

```
Owner: nodeone
```

```
Size: 825930 blocks (403 MB)
```

有关优化的重新同步的更多信息，请参见第 96 页中的“优化的重新同步”。

基于应用程序的恢复和定向镜像读取

要在 Solaris Volume Manager for Sun Cluster 中优化数据恢复，诸如 Oracle Real Application Clusters 的应用程序需要具备管理和控制数据恢复的能力。使用应用程序控制恢复，可以提高恢复的性能。ioctl DKIOGETVOLCAP、DKIOSETVOLCAP 和 DKIODMR 为群集环境中的应用程序数据管理恢复提供了支持。这些 ioctl 为应用程序提供了以下功能：

- 基于应用程序的恢复 (Application Based Recovery, ABR) — 允许应用程序控制镜像卷上的数据恢复
- 定向镜像读取 — 允许应用程序将读取定向到特定子镜像并确定数据的状态

有关用于基于应用程序的数据管理恢复的 ioctl 的更多信息，请参见 `dkio(7I)` 手册页。

只有在基于应用程序的恢复过程中，将重新同步选项设置为基于应用程序的恢复的 RAID-1 卷才具有镜像属主。`metastat -s diskset-name` 命令的以下输出显示了常规状态下的 RAID-1 卷。将重新同步选项设置为基于应用程序的恢复。没有镜像属主。

```
# metastat -s red

red/d24: Mirror

    Submirror 0: red/d20

        State: Okay

    Submirror 1: red/d21

        State: Okay

    Pass: 1

    Read option: roundrobin (default)

    Write option: parallel (default)

    Resync option: application based

    Owner: None

    Size: 825930 blocks (403 MB)
```


配置和使用 Solaris Volume Manager（方案）

在《Solaris Volume Manager 管理指南》中，只要可能，就会为单个存储配置举出关联的示例。本章介绍了这些示例中使用的方案，还提供有关在后续章节中使用的初始存储配置的详细信息。

本章包含以下信息：

- 第 59 页中的“方案背景信息”
- 第 60 页中的“最终的 Solaris Volume Manager 配置”

方案背景信息

在本书中，方案和许多示例都与单个配置有关。尽管此配置很小（以便于简化文档），但在概念上可延伸到更大的存储环境。

硬件配置

硬件系统按以下方式配置：

- 有三个互相分离的控制器（c0 - IDE、c1 - SCSI 和 c2 - SCSI）。
- 每个 SCSI 控制器都连接至包含六个内部 9 GB 磁盘（c1t1 到 c1t6 以及 c2t1 到 c2t6）的 MultiPack。这样做的目的是创建镜像配置。
- 每个控制器/终结符对 (cntn) 都有 8.49 GB 的可用存储空间。
- 根 (/) 驱动器 c0t0d0 上的存储空间被分成六个分区。

下图所示的配置有助于理解以上的配置。

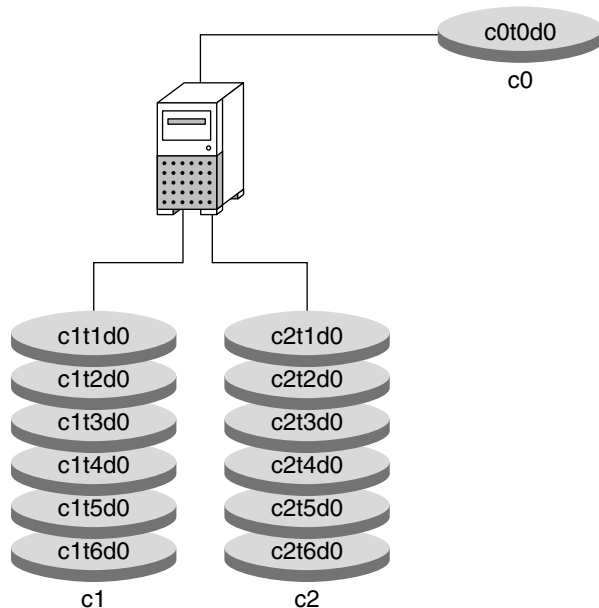


图 5-1 基本硬件存储方案图

初始物理存储配置

以下是配置 Solaris Volume Manager 前的存储配置：

- SCSI 控制器/终结符对 (*cntn*) 大约有 20 GB 的存储空间。
- 每个磁盘（例如 *c1t1d0*）上的存储空间都被分成七个分区（*cntnd0s0* 到 *cntnd0s6*）。

要对磁盘进行分区，请按照《系统管理指南：设备和文件系统》中的“格式化磁盘”中说明的过程执行操作。

最终的 Solaris Volume Manager 配置

本书针对特定任务提供了特定的方案。因此，您可以更好地理解本书中的示例，但最终配置大体上与 `metastat -p` 命令显示的以下内容相同：

```
[root@lexicon:~]$ metastat -p

d50 -r c1t4d0s5 c1t5d0s5 c2t4d0s5 c2t5d0s5 c1t1d0s5 c2t1d0s5 -k -i 32b

d1 1 1 c1t2d0s3
```

```

d2 1 1 c2t2d0s3

d12 1 1 c1t1d0s0

d13 1 1 c2t1d0s0

d16 1 1 c1t1d0s1

d17 1 1 c2t1d0s1

d25 2 2 c1t1d0s3 c2t1d0s3 -i 32b \
    1 c0t0d0s3

d31 1 2 c1t4d0s4 c2t4d0s4 -i 8192b

d80 -p d70 -o 1 -b 2097152

d81 -p d70 -o 2097154 -b 2097152

d82 -p d70 -o 4194307 -b 2097152

d83 -p d70 -o 6291460 -b 2097152

d84 -p d70 -o 8388613 -b 2097152

d85 -p d70 -o 10485766 -b 2097152

d70 -m d71 d72 1

d71 3 1 c1t3d0s3 \
    1 c1t3d0s4 \
    1 c1t3d0s5

d72 3 1 c2t3d0s3 \
    1 c2t3d0s4 \
    1 c2t3d0s5

d123 -p c1t3d0s6 -o 1 -b 204800

d124 -p c1t3d0s6 -o 204802 -b 204800

d125 -p c1t3d0s6 -o 409603 -b 204800

```

```
d126 -p c1t3d0s7 -o 3592 -b 20480
```

```
d127 -p c2t3d0s7 -o 3592 -b 1433600
```

```
hsp010
```

```
hsp014 c1t2d0s1 c2t2d0s1
```

```
hsp050 c1t2d0s5 c2t2d0s5
```

```
hsp070 c1t2d0s4 c2t2d0s4
```

有关 `-p` 选项的更多信息，请参见 `metastat(1M)` 命令。

状态数据库（概述）

本章提供有关状态数据库副本的概念信息。有关执行相关任务的信息，请参见第 7 章。

本章包含以下信息：

- 第 63 页中的 “关于 Solaris Volume Manager 状态数据库和副本”
- 第 64 页中的 “了解多数一致算法”
- 第 66 页中的 “处理状态数据库副本错误”
- 第 67 页中的 “方案一状态数据库副本”

关于 Solaris Volume Manager 状态数据库和副本

Solaris Volume Manager 状态数据库包含所有卷、热备件和磁盘集的配置和状态信息。Solaris Volume Manager 可维护状态数据库的多个副本，以提供冗余并防止数据库在系统崩溃过程中损坏（最多只会有一个数据库副本损坏）。

状态数据库副本可以确保状态数据库中的数据始终有效。当状态数据库被更新时，每个状态数据库副本也被更新。一次只会对一个副本进行更新（以防止系统崩溃的情况下所有更新副本都会损坏）。

如果系统丢失了状态数据库副本，则 Solaris Volume Manager 必须确定哪个状态数据库副本仍然包含有效数据。Solaris Volume Manager 通过使用**多数一致算法**来确定此信息。在此算法中，只有当多数（半数 + 1 个）状态数据库副本可用并一致时，才会认为其中的任何状态数据库副本是有效的。由于多数一致算法的要求，设置磁盘配置时，必须创建至少三个状态数据库副本。只要三个状态数据库副本中至少有两个可用，就可以取得一致。

在引导过程中，Solaris Volume Manager 将忽略损坏的状态数据库副本。在某些情况下，Solaris Volume Manager 会尝试重写损坏的状态数据库副本。否则，在修复这些状态数据库副本之前，会将其忽略。如果状态数据库副本由于其基础片遇到错误而损坏，则需要修复或替换该片，然后再启用副本。



注意 – 请勿将状态数据库副本放置在通过光纤连接系统的存储器、SAN 或未与系统直接连接的其他存储器中。否则可能无法引导 Solaris Volume Manager。副本所在的存储设备在引导过程中必须与传统 SCSI 驱动器或 IDE 驱动器在同一时间可用。

如果丢失所有的状态数据库副本，则理论上可能会丢失存储在 Solaris Volume Manager 卷中的所有数据。出于此原因，最好是在单独的驱动器上跨多个控制器创建足够的状态数据库副本，以防止灾难性故障。另外，保存初始的 Solaris Volume Manager 配置信息以及磁盘分区信息也是明智之举。

有关如何向系统中添加其他状态数据库副本的信息，请参见第 7 章。有关丢失状态数据库副本时如何进行恢复的信息，请参见第 310 页中的“从状态数据库副本故障中恢复”。

状态数据库副本还用于 RAID-1 卷重新同步方面。与镜像数目相比，如果状态数据库副本相对过少，则可能会导致副本 I/O 影响 RAID-1 卷的性能。也即是说，如果有大量镜像，请确保每个 RAID-1 卷至少有两个状态数据库副本，每个磁盘集最多有 50 个副本。

缺省情况下，卷（本地集）和磁盘集的每个状态数据库副本都占用 4 MB（8192 个磁盘扇区）的磁盘存储空间。多属主磁盘集的状态数据库副本的缺省大小为 16 MB。

副本可以存储在以下设备中：

- 专用本地磁盘分区
- 属于卷的本地分区
- 属于 UFS 日志记录设备的本地分区

副本不能存储在根 (/)、swap 或 /usr 片上，也不能存储在包含现有文件系统或数据的片上。副本存储后，可以将卷或文件系统放置在同一个片上。

了解多数一致算法

复制的数据库的固有问题是可能很难确定哪个数据库包含有效并且正确的数据。为了解决此问题，Solaris Volume Manager 使用多数一致算法。此算法要求大多数数据库副本彼此一致，然后才能声明其中任一副本有效。此算法还要求至少存在三个已创建的初始副本。只要三个副本中至少有两个可用，就可以取得一致。如果仅存在一个副本并且系统已崩溃，则可能会丢失所有的卷配置数据。

为了保护数据，只有当所有状态数据库副本中有一半可用时，Solaris Volume Manager 才会运行。因此，该算法可确保不会损坏数据。

多数一致算法可提供以下功能：

- 如果至少有一半状态数据库副本可用，则系统将运行。

- 如果少于一半的状态数据库副本可用，则系统将崩溃。
- 仅当全部状态数据库副本中的多数（半数 + 1）可用时，系统才能重新引导至多用户模式。

如果可用的状态数据库副本数量不足，则必须将系统引导至单用户模式，并且删除足够的已损坏副本或缺失副本以达到定额。请参见第 310 页中的“[如何从状态数据库副本不足中恢复](#)”。

注 - 如果状态数据库副本总数为奇数，则 Solaris Volume Manager 会将该数字除以 2，然后将结果向下舍入到最接近的整数，再加上 1（一）以计算表示多数的数字。例如，在包含 7 个副本的系统上，表示多数的数字是 4（7 除以 2 为 3.5，向下舍入后为 3，再加 1 为 4）。

管理状态数据库副本

- 缺省情况下，状态数据库副本的大小为 4 MB 或 8192 个块。应在专用片上创建每个副本大小至少为 4 MB 的状态数据库副本。由于磁盘片可能不会这么小，因此可能需要调整片的大小才能存放状态数据库副本。有关调整片的大小的信息，请参见《系统管理指南：设备和文件系统》中的第 12 章“管理磁盘（任务）”。
- 要避免单点故障，请将状态数据库副本分布于片、驱动器和控制器上。一般希望在单个组件发生故障时大部分副本完好。如果丢失了副本（例如，由于设备故障），则运行 Solaris Volume Manager 或重新引导系统时可能会出现问题。Solaris Volume Manager 要求至少有一半副本可用时才能运行，但要求有大多数（半数 + 1 个）副本可用才能重新引导至多用户模式。

建议至少要有 3 个状态数据库副本，每个 Solaris Volume Manager 磁盘集最多有 50 个副本。建议遵循以下原则：

- 对于只有一个驱动器的系统：一个片上放置全部三个副本。
- 对于具有两个到四个驱动器的系统：一个驱动器上放置两个副本。
- 对于具有五个或更多驱动器的系统：一个驱动器上放置一个副本。
- 如果存在多个控制器，则应将副本尽可能平均分布于所有控制器中。此策略可在某个控制器出现故障时提供冗余性，并且有助于平衡负载。如果控制器上存在多个磁盘，则每个控制器上至少有两个磁盘应存储副本。
- 如有必要，可以在将用作 RAID-0、RAID-1 或 RAID-5 卷的一部分的片或软分区中创建状态数据库副本。向卷中添加片之前，必须创建副本。Solaris Volume Manager 可为状态数据库副本保留片的起始部分。

如果将状态数据库副本放置在作为卷的一部分的片上，则卷的容量会减小，这是由于副本会占用一定空间。副本使用的空间会逐渐累积，直至达到下一个柱面边界为止。卷将跳过此空间。

- RAID-1 卷用于小规模随机 I/O（如数据库中的随机 I/O）。要获得最佳性能，请在未连接到 RAID-1 卷的片上（最好位于单独的磁盘和控制器上）的每个 RAID-1 卷中至少放置两个附加副本。

- 不能在现有文件系统，或根 (/)、/usr 和 swap 文件系统上创建状态数据库副本。如有必要，可通过分配 swap 的空间来创建新片（假定片名称可用）。然后，将状态数据库副本放置在此新片上。
- 可以在未使用的片上创建状态数据库副本。
- 可以随时向系统中添加附加的状态数据库副本。附加的状态数据库副本有助于确保 Solaris Volume Manager 的可用性。



注意 – 如果已从 Solstice DiskSuite 产品升级到 Solaris Volume Manager，并且使状态数据库副本与文件系统或逻辑卷共享片（与单独的片相对），请不要删除现有副本并将其替换为同一位置中的新副本。

Solaris Volume Manager 中状态数据库的缺省大小为 8192 个块，而 Solstice DiskSuite 产品中的缺省大小为 1034 个块。删除在 Solstice DiskSuite 产品中创建的缺省大小的状态数据库副本，然后使用 Solaris Volume Manager 添加新的缺省大小的副本时，请务必小心谨慎。这会覆写占用共享片其余空间的任何文件系统的前 7158 个块，从而会损坏数据。

处理状态数据库副本错误

状态数据库副本出现故障时，如果剩余副本中至少有一半可用，则系统会继续运行。如果少于一半的副本可用，则系统将崩溃。

可用的副本至少要比半数多一个，系统才能重新引导至多用户模式。如果可用副本少于多数，则必须将系统重新引导至单用户模式，并删除不可用的副本（通过使用 `metadb` 命令）。

例如，假定您有四个副本。只要两个副本（总数的一半）可用，系统就会继续运行。但是，要重新引导系统，则必须有三个副本（总数的一半 + 1）可用。

在双磁盘配置中，应始终在每个磁盘上至少创建两个副本。例如，假定配置中包含两个磁盘，但仅创建了三个副本（第一个磁盘上有两个副本，另一个磁盘上有一个副本）。如果包含两个副本的磁盘出现了故障，则系统会崩溃，这是因为剩余的磁盘仅有一个副本。也即是说，少于副本总数的一半。

注 – 如果在双磁盘配置中的每一个磁盘上都创建两个副本，则当其中一个磁盘出现故障时，Solaris Volume Manager 仍可继续运行。但是，由于可用的副本必须多于总数的一半时系统才能重新引导，因此无法重新引导系统。

如果包含状态数据库副本的片出现故障，则配置中的其余部分应当仍然继续操作。Solaris Volume Manager 可在引导过程中找到有效的状态数据库（只要可用的有效状态数据库副本比半数多一个）。

手动修复或启用状态数据库副本时，Solaris Volume Manager 将使用有效数据对这些副本进行更新。

方案－状态数据库副本

状态数据库副本可提供有关总体 Solaris Volume Manager 配置的冗余数据。以下示例基于第 5 章中所提供的方案中的样例系统。本示例介绍如何分布状态数据库副本以提供足够的冗余性。

样例系统具有一个内部 IDE 控制器和驱动器，还有两个 SCSI 控制器。每个 SCSI 控制器都连接有六个磁盘。可以使用三个控制器对系统进行配置，从而避免任何单点故障。任何仅有两个控制器的系统都无法避免与 Solaris Volume Manager 有关的单点故障。通过将副本平均分布于所有三个控制器以及每个控制器中的至少一个磁盘上（如果可能，分布于两个磁盘），系统可以经受任何单个硬件故障。

在最低配置中，可以将单个状态数据库副本放置在根磁盘的片 7 上，然后将其他副本放置在其他两个控制器中每一个上的某个磁盘的片 7 上。为了帮助应对可能出现的众所周知的远程介质故障，请向根磁盘中添加另一个副本，这样，位于每个控制器上的两个不同磁盘中的两个副本（总共为六个副本）便可提供足够的安全性。

要提供更高的安全性，请添加 12 个附加副本，将其平均分布于两个镜像的每一面上的 6 个磁盘中。此配置将产生总共 18 个副本，其中根磁盘上有 2 个副本，每个 SCSI 控制器上有 8 个副本，这些副本分布于每个控制器上的磁盘中。

状态数据库（任务）

本章提供有关执行与 Solaris Volume Manager 状态数据库副本相关联的任务的信息。有关这些任务中所涉及概念的信息，请参见第 6 章。

状态数据库副本（任务列表）

以下任务列表列出了管理 Solaris Volume Manager 状态数据库副本所需的过程。

任务	说明	参考
创建状态数据库副本	使用 Solaris Volume Manager GUI 或 <code>metadb -a</code> 命令创建状态数据库副本。	第 70 页中的“如何创建状态数据库副本”
检查状态数据库副本的状态	使用 Solaris Volume Manager GUI 或 <code>metadb</code> 命令检查现有副本的状态。	第 72 页中的“如何检查状态数据库副本的状态”
删除状态数据库副本	使用 Solaris Volume Manager GUI 或 <code>metadb -d</code> 命令删除状态数据库副本。	第 73 页中的“如何删除状态数据库副本”

创建状态数据库副本



注意 – 如果您从 Solstice DiskSuite 产品升级到 Solaris Volume Manager，而且状态数据库副本与文件系统或逻辑卷共享片（与使用单独的片相反），请不要删除现有的副本或在同一个位置用新的缺省副本来替换它。

Solaris Volume Manager 中状态数据库的缺省大小为 8192 个块，而 Solstice DiskSuite 产品中的缺省大小是 1034 个块。删除在 Solstice DiskSuite 产品中创建的缺省大小的状态数据库副本，然后使用 Solaris Volume Manager 添加新的缺省大小的副本时，请务必小心谨慎。这会覆写占用共享片其余空间的任何文件系统的前 7158 个块，从而会销毁数据。



注意 – 请勿将状态数据库副本放置在通过光纤连接系统的存储器、SAN 或未与系统直接连接的其他存储器中。否则可能无法引导 Solaris Volume Manager。副本所在的存储设备必须与传统的 SCSI 或 IDE 驱动器在引导过程中同时可用。

▼ 如何创建状态数据库副本

开始之前 请查看第 45 页中的“创建 Solaris Volume Manager 组件的先决条件”。

- 1 成为超级用户。
- 2 要创建状态数据库副本，请使用以下方法之一：
 - 从 Solaris Management Console 中增强的存储工具中，打开“状态数据库副本”节点。选择“操作”⇒“创建副本”并按照屏幕上的说明操作。有关更多信息，请参见联机帮助。
 - 使用 `metadb` 命令的以下格式。请参见 `metadb(1M)`。

```
# metadb -a -c number -l length-of-replica -f ctds-of-slice
```

-a 指定要添加或创建状态数据库副本。

-f 指定要强制执行该操作，即使没有副本存在也是如此。使用 -f 可强制创建初始副本。

-c number 指定要添加到给定片中的副本数量。

-l length-of-replica 指定新副本的大小（用块表示）。缺省大小是 8192。此大小应适合几乎所有的配置（包括那些具有数千个逻辑卷的配置）。

ctds-of-slice 指定将用来存放副本的部件的名称。

注 - 在命令行上输入的不带任何选项的 `metadb` 命令可报告所有状态数据库副本的状态。

示例 7-1 创建第一个状态数据库副本

```
# metadb -a -f c0t0d0s7

# metadb

      flags      first blk      block count

...

      a          u          16          8192          /dev/dsk/c0t0d0s7
```

必须使用 `-f` 和 `-a` 选项来创建第一个状态数据库副本。`-a` 选项用来向系统中添加状态数据库副本，`-f` 选项可强制创建第一个副本（在向系统中添加补充副本时，可以省略该选项）。

示例 7-2 向同一个片中添加两个状态数据库副本

```
# metadb -a -c 2 c1t3d0s1

# metadb

      flags      first blk      block count

...

      a          u          16          8192          /dev/dsk/c1t3d0s1

      a          u          8208         8192          /dev/dsk/c1t3d0s1
```

`-a` 选项向系统中添加状态数据库副本，`-c 2` 选项将两个副本放到指定的片上。`metadb` 命令检查副本是否处于活动状态，活动状态由 `metadb` 命令输出中的 `a` 标志来指示。

示例 7-3 添加指定大小的状态数据库副本

如果要替换现有的状态数据库副本，则可能需要指定副本的大小。特别是在现有的状态数据库副本（可能位于从 Solstice DiskSuite 产品升级的系统上）与文件系统共享一个片时，必须将现有的副本替换为相同大小的其他副本，或者在另一个位置中添加新副本。

```
# metadb -a -c 3 -l 1034 c0t0d0s7

# metadb

      flags      first blk      block count
...
      a      u      16      1034      /dev/dsk/c0t0d0s7
      a      u      1050      1034      /dev/dsk/c0t0d0s7
      a      u      2084      1034      /dev/dsk/c0t0d0s7
```

-a 选项向系统中添加状态数据库副本，-l 选项用于指定要添加的副本的长度（用块表示）。

维护状态数据库副本

▼ 如何检查状态数据库副本的状态

- 1 成为超级用户。
- 2 要检查状态数据库副本的状态，请使用以下方法之一：
 - 从 Solaris Management Console 中增强的存储工具中，打开“状态数据库副本”节点以查看现有的全部状态数据库副本。有关更多信息，请参见联机帮助。
 - 使用 metadb 命令查看状态数据库副本的状态。添加 -i 选项以显示状态标志的说明，如下示例中所示。请参见 metadb(1M)。

示例 7-4 检查所有状态数据库副本的状态

```
# metadb -i

      flags      first blk      block count
...
      a m p luo      16      8192      /dev/dsk/c0t0d0s7
      a      p luo      8208      8192      /dev/dsk/c0t0d0s7
      a      p luo      16400      8192      /dev/dsk/c0t0d0s7
```


a	p	luo	16	8192	/dev/dsk/clt3d0s1
W	p	l	16	8192	/dev/dsk/c2t3d0s1
a	p	luo	16	8192	/dev/dsk/clt1d0s3
a	p	luo	8208	8192	/dev/dsk/clt1d0s3
a	p	luo	16400	8192	/dev/dsk/clt1d0s3

- r - replica does not have device relocation information
- o - replica active prior to last mddb configuration change
- u - replica is up to date
- l - locator for this replica was read successfully
- c - replica's location was in /etc/lvm/mddb.cf
- p - replica's location was patched in kernel
- m - replica is master, this is replica selected as input
- W - replica has device write errors
- a - replica is active, commits are occurring to this replica
- M - replica had problem with master blocks
- D - replica had problem with data blocks
- F - replica had format problems
- S - replica is too small to hold current data base
- R - replica had device read errors

所有标志的说明后跟相应的状态。设备名称前面的字符表示状态。大写字母表示处于问题状态，小写字母表示处于“正常”状态。

▼ 如何删除状态数据库副本

可能需要删除状态数据库副本以维护 Solaris Volume Manager 配置。例如，如果要替换磁盘驱动器，则需要在删除驱动器之前先删除状态数据库副本。否则，Solaris Volume Manager 将报告磁盘驱动器有错误。

1 成为超级用户。

2 要删除状态数据库副本，请使用以下方法之一：

- 从 Solaris Management Console 中增强的存储工具中，打开“状态数据库副本”节点以查看现有的全部状态数据库副本。选择要删除的副本，然后选择“编辑”⇒“删除”将它们删除。有关更多信息，请参见联机帮助。
- 使用 `metadb` 命令的以下格式：

```
# metadb -d -f ctds-of-slice
```

-d 指定要删除状态数据库副本。

-f 指定要强制执行该操作，即使没有副本存在也是如此。

ctds-of-slice 指定包含副本的部件的名称。

请注意，需要指定要删除其中的状态数据库副本的每个片。有关更多信息，请参见 `metadb(1M)` 手册页。

示例 7-5 删除状态数据库副本

```
# metadb -d -f c0t0d0s7
```

此示例说明如何从片中删除最后一个副本。

必须添加 `-f` 选项才能强制删除系统上的最后一个副本。

RAID-0（条带化和串联）卷（概述）

本章介绍 Solaris Volume Manager 中可用的 RAID-0（条带化和串联）卷。有关相关任务的信息，请参见第 9 章。

本章提供以下信息：

- 第 75 页中的“RAID-0 卷概述”
- 第 81 页中的“创建 RAID-0 卷的背景信息”
- 第 81 页中的“方案—RAID-0 卷”

RAID-0 卷概述

RAID-0 卷由片或软分区组成。使用这些卷，可以扩展磁盘存储容量。可以直接使用这些卷，也可以将它们用作 RAID-1（镜像）卷和软分区的生成块。有三种 RAID-0 卷：

- 条带化卷
- 串联卷
- 串联条带化卷

注 - 组件是指其他逻辑卷中使用的从片到软分区的任何设备。

条带化卷将数据平均分布于卷中的所有组件中，而串联卷先将数据写入第一个可用组件中，直到将该组件写满后才移至下一个可用组件。串联条带化卷仅仅是通过添加附件组件从其原始配置扩展后的条带化卷。

使用 RAID-0 卷，可以快速且简单地扩展磁盘存储容量。缺点是这些卷不提供任何数据冗余，这与 RAID-1 或 RAID-5 卷不同。如果 RAID-0 卷中单个组件出现故障，则数据将丢失。

对于条带化卷中的顺序 I/O 操作，Solaris Volume Manager 将读取第一个组件上的块（称为**交错**）区段中的所有块，然后读取第二个组件上的块区段中的所有块，依此类推。

对于串联卷上的顺序 I/O 操作，Solaris Volume Manager 将读取第一个组件上的所有块，然后读取第二个组件的所有块，依此类推。

在串联卷和条带化卷上，所有的 I/O 操作都是并行发生的。

可以对任何文件系统使用包含单个片的 RAID-0 卷。

可以对以下文件系统以外的任何文件系统使用包含多个组件的 RAID-0 卷：

- 根 (/)
- /usr
- swap
- /var
- /opt
- 升级或安装操作系统期间访问的任何文件系统

注 - 镜像根 (/)、/usr、swap、/var 或 /opt 时，可以将文件系统置于作为子镜像的单向串联或条带（单个片的串联）中。此单向串联由另一个子镜像（也必须为串联）来进行镜像。

RAID-0（条带化）卷

RAID-0（条带化）卷是跨一个或多个组件安排数据存储的卷。条带化时会跨越两个或更多组件交替存储大小相等的数据区段，形成一个逻辑存储单元。这些区段循环交换，使得组合的空间由各组件交替组成，效果上就像洗牌一样。

注 - 要增大条带化卷的容量，需要生成串联条带化卷。请参见第 79 页中的“RAID-0（串联条带化）卷”。

条带化后，多个控制器可以同时访问数据，这也称为**并行访问**。并行访问可以增大 I/O 吞吐量，因为卷中的所有磁盘在大多数时间里都在忙于为 I/O 请求提供服务。

不能将现有文件系统直接转换为条带化系统。要将现有文件系统置于条带化卷上，必须备份文件系统、创建卷，然后将文件系统恢复为条带化卷。

RAID-0（条带化）卷的交错值

交错值是指条带化卷上的逻辑数据区段的大小（以 KB、MB 为单位）或块数。可以根据应用程序使用不同的交错值提高配置的性能。性能的提高是通过让多个磁盘装备管理 I/O 请求来实现。当 I/O 请求的大小大于交错大小时，可能会获得更好的性能。

注 – RAID-5 卷也使用交错值。有关更多信息，请参见第 161 页中的“RAID-5 卷的概述”。

创建条带化卷时，可以设置交错值，也可以使用 Solaris Volume Manager 缺省交错值 16 KB。创建条带化卷后，不能更改交错值。但是，可以备份条带化卷上的数据、删除该卷、使用新的交错值创建新的条带化卷，然后恢复数据。

方案 – RAID-0（条带化）卷

图 8-1 显示使用三个组件（片）生成的条带化卷。该图还说明了如何根据交错大小并使用循环技术将数据写入卷组件。

当 Solaris Volume Manager 将数据写入条带化卷的组件时，会将交错宽度的数据块写入磁盘 A（交错 1）、磁盘 B（交错 2）、磁盘 C（交错 3）。随后，Solaris Volume Manager 将重复该模式向磁盘 A（交错 4）、磁盘 B（交错 5）、磁盘 C（交错 6）写入数据，依此类推。

交错值设置每次写入片的数据的大小。条带化卷的总容量等于组件数目乘以最小组件的大小。（如果以下示例中的每一片都是 2 GB，则卷将等于 6 GB。）

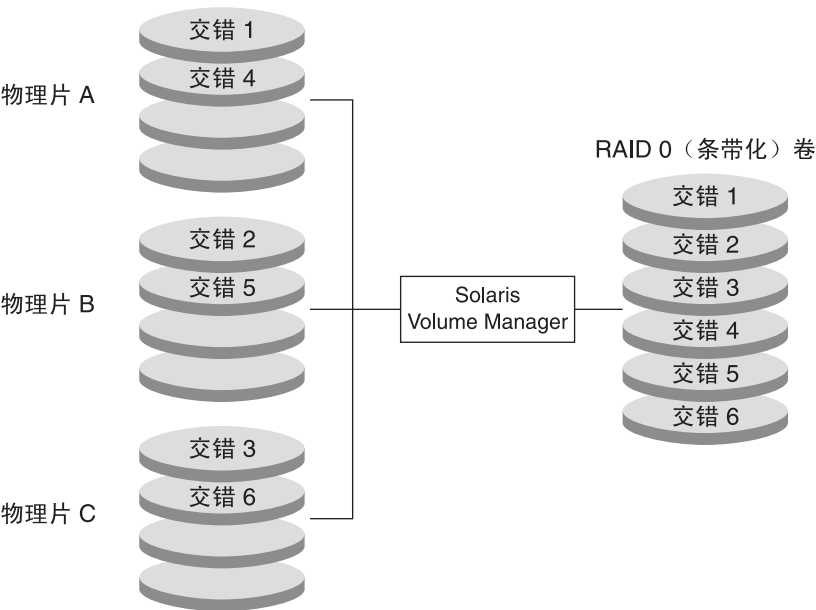


图 8-1 RAID-0（条带化）卷示例

RAID-0（串联）卷

RAID0-0（串联）卷中，数据在各组件上连续无间隔地进行存储，形成一个逻辑存储单元。

使用串联卷，通过将多个组件的容量组合在一起，可以获得更多存储容量。随着存储需求的不断增加，可以向串联卷中添加更多组件。

使用串联卷，可以在联机情况下动态扩展存储容量和文件系统大小。此外，使用串联卷也可以在其他组件当前处于活动状态时添加组件。

串联卷还可以在无需关闭系统的情况下，扩展任何活动的、已挂载的 UFS 文件系统。一般情况下，串联卷的总容量等于卷中所有组件的容量总和。如果串联卷包含带有状态数据库副本的片，则卷的总容量将为各组件容量之和减去为该副本保留的空间后所得的结果。

可以使用单个组件来创建串联卷。以后需要更多存储空间时，可以向卷中添加更多组件。

注 – 镜像根 (/)、swap、/usr、/opt 或 /var 文件系统时，必须使用串联卷来封装它们。

方案 – RAID-0（串联）卷

图 8-2 说明使用三个组件（片）生成的串联卷。该图还说明了如何根据交错大小将数据按顺序写入卷组件，进而写入每个片。

数据块的写入模式是跨各个组件按顺序（从片 A 开始）写入。可以将片 A 想象为包含逻辑数据块 1 - 4。磁盘 B 包含逻辑数据块 5 - 8。驱动器 C 包含逻辑数据块 9 - 12。卷的总容量将为三个片的组合容量。如果每片为 2 GB，则卷将具有 6 GB 的总容量。

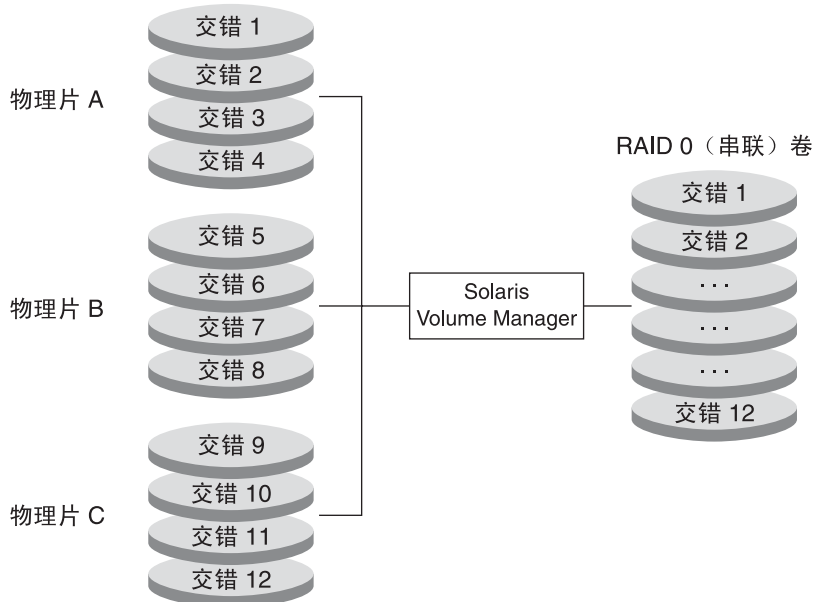


图 8-2 RAID-0（串联）卷示例

RAID-0（串联条带化）卷

RAID-0（串联条带化）卷是通过添加附加组件（条带）扩展的条带。

要在条带级别上为串联条带化卷设置交错值，请使用 Solaris Management Console 中增强的存储工具 或 `metattach -i` 命令。串联条带化卷中的每个条带都可以具有各自的交错值。从头创建串联条带化卷时，如果未针对特定条带指定交错值，则该条带将沿用添加到卷中的前一个条带的交错值。

示例—RAID-0（串联条带化）卷

图 8-3 说明了串联了三个条带的串联条带化卷。

第一个条带包含三个片，片 A - C，交错值为 16 KB。第二个条带包含两个片，片 D 和 E，且使用交错值 32 KB。最后一个条带包含两个片，片 F 和 G。由于没有为第三个条带指定任何交错值，因此该条带将沿用在它之前添加的条带的值，在本例中为 32 KB。系统会将连续的数据块添加到第一个条带中，直到该条带不再有空间为止。随后，会将数据块添加到第二个条带中。当此条带不再有空间时，就会将数据块添加到第三个条带中。在每个条带内，数据块将根据指定的交错值交错分布。

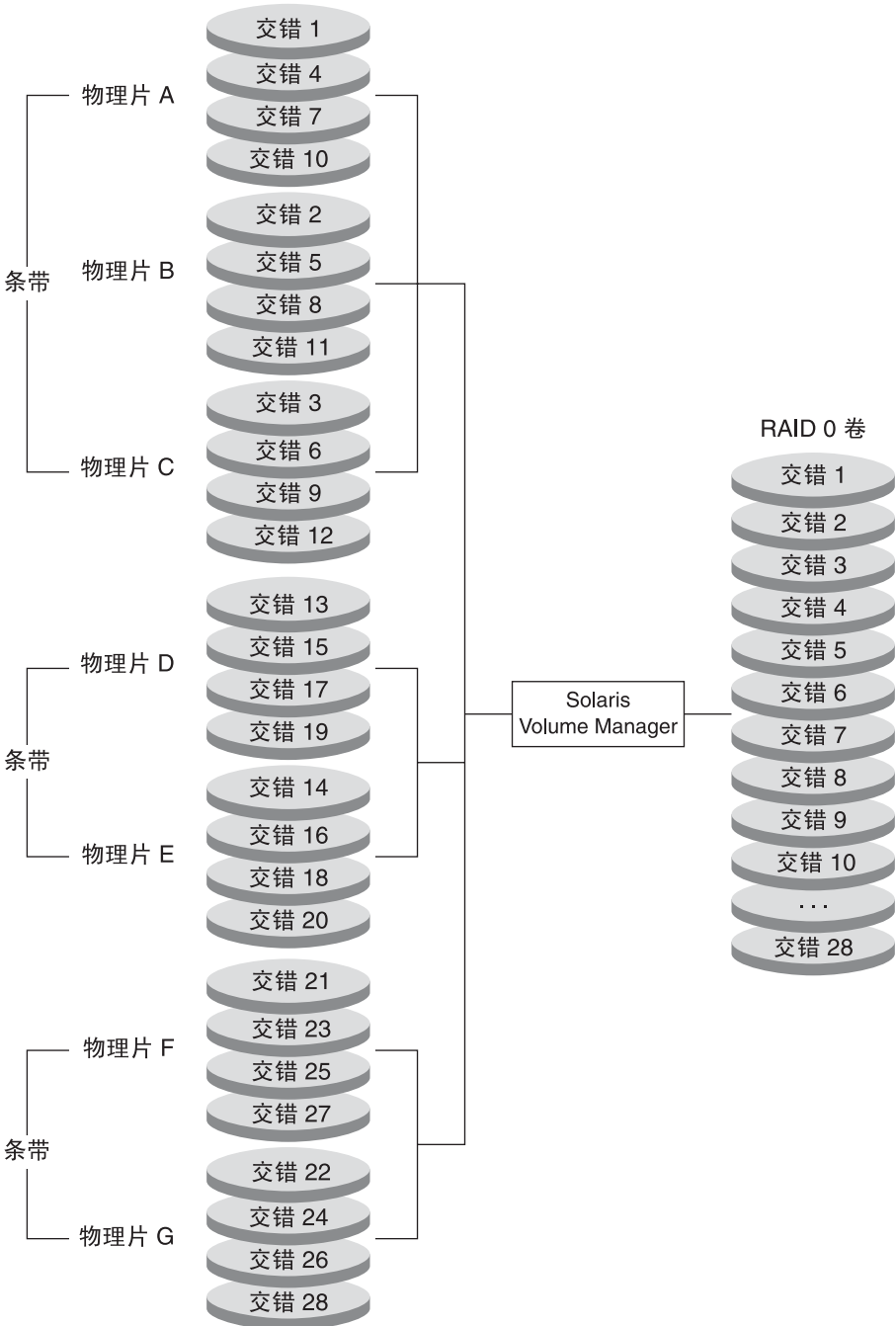


图 8-3 RAID-0（串联条带化）卷示例

创建 RAID-0 卷的背景信息

RAID-0 卷要求

使用 RAID-0 卷时，请考虑以下几点：

- 使用分别位于不同控制器上的组件，以增加可同时执行的读取和写入操作的数目。
- 请勿从现有的文件系统或数据创建条带，这样做会销毁数据。请改用串联。（可以使用现有数据创建条带，但必须将数据转储后再将数据恢复到卷。）
- 对条带使用大小相同的磁盘组件。大小不同的条带化组件将导致磁盘空间浪费。
- 设置可更好地匹配系统或应用程序发出的 I/O 请求的条带交错值。
- 由于条带化卷或串联卷不包含数据副本，因此当这类卷出现组件故障时，必须替换该组件、重新创建条带化卷或串联卷，并从备份中恢复数据。
- 重新创建条带或串联时，请使用大小至少与出现故障的组件相等的替换组件。

RAID-0 卷指南

- 串联使用的 CPU 周期少于条带化，且非常适用于少量随机 I/O 和均匀 I/O 分布。
- 如果可能，请将条带化卷或串联卷的组件分布于不同的控制器和总线中。使用分别位于不同控制器上的条带，可以增加可同时执行的读取和写入操作的数目。
- 如果在出现故障的控制器上定义了条带且系统上有另一个可用的控制器，则通过将磁盘移动到新的控制器并重新定义条带，可以将该条带“移动”到新控制器。
- 条带数目：考虑条带化的另一个方面是首先确定性能要求。例如，您可能需要为选定应用程序提供 10.4 MB/秒的性能，而每个磁盘的传送速度大约能达到 4 MB/秒。根据以下公式，确定需要跨多少个磁盘轴进行条带化：

$$10.4 \text{ Mbyte/sec} / 4 \text{ Mbyte/sec} = 2.6$$

由此可知，需要三个能并行执行 I/O 操作的磁盘。

方案—RAID-0 卷

RAID-0 卷提供用于创建较复杂存储配置或生成镜像的基础生成块。以下示例（利用第 5 章中说明的方案）介绍了 RAID-0 卷如何提供较大的存储空间，以及如何使用这类卷来构造现有文件系统（包括根 (/)）的镜像。

该方案中的样例系统具有相对较小 (9 GB) 的磁盘集合，但特定应用程序可能需要较大的存储空间。要创建较大的空间（并提高性能），可以创建跨多个磁盘的条带。例如，以下磁盘（c1t1d0、c1t2d0、c1t3d0、c2t1d0、c2t2d0 和 c2t3d0）中的每一个都可以使用跨整个磁盘的片 0 进行格式化。之后，包括某个控制器中的所有三个磁盘的条

带就可以提供大约 27 GB 的存储空间，且允许更快速的访问。第二个控制器中的第二个条带可用于冗余操作，如第 11 章中所述，具体信息请参见第 101 页中的“方案—RAID-1 卷（镜像）”。

RAID-0（条带化和串联）卷（任务）

本章介绍有关与 RAID-0 卷相关的任务的信息。有关相关概念的信息，请参见第 8 章。

RAID-0 卷（任务列表）

以下任务列表列出了管理 Solaris Volume Manager RAID-0 卷所需的过程。

任务	说明	参考
创建 RAID-0（条带化）卷	使用 Solaris Volume Manager GUI 或 <code>metainit</code> 命令创建新卷。	第 84 页中的“如何创建 RAID-0（条带化）卷”
创建 RAID-0（串联）卷	使用 Solaris Volume Manager GUI 或 <code>metainit</code> 命令创建新卷。	第 85 页中的“如何创建 RAID-0（串联）卷”
扩展存储空间	使用 Solaris Volume Manager GUI 或 <code>metainit</code> 命令扩展现有的文件系统。	第 87 页中的“如何扩展现有数据的存储容量”
扩展现有的 RAID-0 卷	使用 Solaris Volume Manager GUI 或 <code>metattach</code> 命令扩展现有的卷。	第 88 页中的“如何扩展现有的 RAID-0 卷”
删除 RAID-0 卷	使用 Solaris Volume Manager GUI 或 <code>metaclear</code> 命令删除卷。	第 90 页中的“如何删除 RAID-0 卷”

创建 RAID-0（条带化）卷



注意 – 请勿从现有的文件系统或数据创建条带，否则会销毁数据。要从现有的数据创建条带，必须备份数据，创建条带化卷，然后将数据恢复到该卷上。



注意 – 如果您希望运行具有 32 位内核的 Solaris 软件，请不要创建大于 1TB 的卷。另外，如果您希望运行版本低于 Solaris 9 4/03 发行版的 Solaris OS，也不要创建大于 1TB 的卷。有关 Solaris Volume Manager 中大卷支持的更多信息，请参见第 45 页中的“[Solaris Volume Manager 中的多 TB 支持的概述](#)”。

▼ 如何创建 RAID-0（条带化）卷

开始之前 请查看第 45 页中的“[创建 Solaris Volume Manager 组件的先决条件](#)”和第 81 页中的“[创建 RAID-0 卷的背景信息](#)”。

► 要创建条带化卷，请使用以下方法之一：

- 从 Solaris Management Console 中增强的存储工具中，打开“卷”节点。选择“操作”⇒“创建卷”，然后按照向导中的说明操作。有关更多信息，请参见联机帮助。
- 使用以下形式的 `metainit` 命令：

```
# metainit volume-name number-of-stripes
```

```
components-per-stripe
```

```
component-names
```

```
[ -i interlace]
```

volume-name 指定要创建的卷的名称。有关对卷进行命名的信息，请参见第 42 页中的“[卷名称](#)”。

number-of-stripes 指定要创建的条带的数量。

components-per-stripe 指定每个条带应包含的组件的数目。

component-names 指定所使用的组件的名称。如果使用多个组件，请用空格分隔它们。

-i interlace 指定要用于条带的交错宽度。交错宽度是一个值，其后面是“k”（表示千字节）、“m”（表示兆字节）或“b”（表示块）。指定的交错不能小于 16 块，也不能大于 100 MB。缺省的交错宽度是 16 KB。

有关更多信息，请参见以下示例和 `metainit(1M)` 手册页。

示例 9-1 创建由三个片组成的 RAID-0（条带化）卷

```
# metainit d20 1 3 c0t1d0s2 c0t2d0s2 c0t3d0s2
```

```
d20: Concat/Stripe is setup
```

以上示例说明条带 **d20** 包含一个条带（由数字 **1** 指示）。该条带由三个片组成（由数字 **3** 指示）。由于未指定交错值，因此该条带使用缺省值 **16 KB**。系统会确认该卷是否已经设置。

示例 9-2 创建由两个片组成且交错值等于 32 的 RAID-0（条带化）卷

```
# metainit d10 1 2 c0t1d0s2 c0t2d0s2 -i 32k
```

```
d10: Concat/Stripe is setup
```

以上示例显示条带 **d10** 包含一个条带（由数字 **1** 指示）。该条带由两个片组成（由数字 **2** 指示）。**-i** 选项将交错值设置为 **32 KB**。（交错值不能小于 **8 KB**，也不能大于 **100 MB**。）系统会验证该卷是否已经设置。

另请参见 要为文件系统准备新创建的条带，请参见《系统管理指南：设备和文件系统》中的第 18 章“创建 UFS、TMPFS 和 LOFS 文件系统（任务）”。某些应用程序（如数据库）不使用文件系统，而是使用原始设备。这些应用程序必须能够以自己的方式来访问原始设备。

创建 RAID-0（串联）卷

▼ 如何创建 RAID-0（串联）卷



注意 – 如果您希望运行具有 32 位内核的 Solaris 软件，请不要创建大于 1TB 的卷。另外，如果您希望运行版本低于 Solaris 9 4/03 发行版的 Solaris OS，请不要创建大于 1TB 的卷。有关 Solaris Volume Manager 中多 TB 卷的更多信息，请参见第 45 页中的[“Solaris Volume Manager 中的多 TB 支持的概述”](#)。

开始之前 请查看第 45 页中的“创建 Solaris Volume Manager 组件的先决条件”和第 81 页中的“创建 RAID-0 卷的背景信息”。

► 要创建串联卷，请使用以下方法之一：

- 从 Solaris Management Console 中增强的存储工具中，打开“卷”节点。选择“操作” ⇒ “创建卷”，然后按照向导中的说明操作。有关更多信息，请参见联机帮助。
- 使用以下形式的 **metainit** 命令：

```
# metainit volume-name number-of-stripes
```

*components-per-stripe**component-names**volume-name*

指定要创建的卷的名称。

number-of-stripes

指定要创建的条带的数量。

components-per-concatenation

指定每个串联应当拥有的组件的数量。

component-names

指定所使用的组件的名称。如果使用多个组件，请用空格分隔它们。

有关更多信息，请参见以下示例和 `metainit(1M)` 手册页。**示例 9-3 创建由一个片组成的串联**

```
# metainit d25 1 1 c0t1d0s2
```

```
d25: Concat/Stripe is setup
```

此示例说明如何创建串联 `d25`。此串联由一个条带组成（由第一个数字 `1` 指示），该条带由一个片组成（由该片之前的第二个数字 `1` 指示）。系统会验证该卷是否已经设置。

此示例说明了可以安全地封装现有数据的串联。

示例 9-4 创建由四个片组成的串联

```
# metainit d40 4 1 c0t1d0s2 1 c0t2d0s2 1 c0t2d0s3 1 c0t2d1s3
```

```
d40: Concat/Stripe is setup
```

此示例说明如何创建串联 `d40`。该串联由四个条带组成（由数字 `4` 指示），每个条带由一个片（由每个片之前的数字 `1` 指示）组成。系统会验证该卷是否已经设置。

另请参见 要为文件系统准备新创建的串联，请参见《系统管理指南：设备和文件系统》中的第 18 章“创建 UFS、TMPFS 和 LOFS 文件系统（任务）”。

扩展存储容量

要向文件系统中添加存储容量，请创建串联卷。要向现有条带中添加存储容量，请创建串联条带化卷。

▼ 如何扩展现有数据的存储容量



注意 – 如果您希望运行具有 32 位内核的 Solaris 软件，请不要创建大于 1TB 的卷。另外，如果您希望运行版本低于 Solaris 9 4/03 发行版的 Solaris OS，请不要创建大于 1TB 的卷。有关 Solaris Volume Manager 中多 TB 卷支持的更多信息，请参见第 45 页中的“[Solaris Volume Manager 中的多 TB 支持的概述](#)”。

开始之前 请查看第 45 页中的“创建 Solaris Volume Manager 组件的先决条件”和第 81 页中的“创建 RAID-0 卷的背景信息”。

1 取消挂载文件系统。

```
# umount /filesystem
```

2 要创建串联，请使用以下方法之一：

- 从 Solaris Management Console 中增强的存储工具中，打开“卷”节点。选择“操作” ⇒ “创建卷”，然后按照向导中的说明操作。有关更多信息，请参见联机帮助。
- 使用以下形式的 `metainit` 命令：

```
# metainit volume-name number-of-stripes
```

```
components-per-stripe
```

```
component-names
```

volume-name 指定要创建的卷的名称。

number-of-stripes 指定要创建的条带的数量。

components-per-stripe 指定每个条带应当拥有的组件的数量。

component-names 指定所使用的组件的名称。如果使用多个组件，请用空格分隔它们。

有关更多信息，请参见 `metainit(1M)` 手册页。

3 编辑 `/etc/vfstab` 文件，以便文件系统可以引用串联的名称。

4 重新挂载文件系统。

```
# mount /filesystem
```

示例 9-5 通过创建串联卷来扩展文件系统

```
# umount /docs
```

```
# metainit d25 2 1 c0t1d0s2 1 c0t2d0s2
```

```
d25: Concat/Stripe is setup
```

（编辑 */etc/vfstab* 文件使该文件系统引用卷 *d25* 而非片 *c0t1d0s2*）

```
# mount /docs
```

此示例说明如何从两个片 */dev/dsk/c0t1d0s2*（该片包含挂载在 */docs* 上的文件系统）和 */dev/dsk/c0t2d0s2* 创建串联 *d25*。必须首先取消文件系统的挂载。请注意，*metainit* 命令中的第一个片必须是包含该文件系统的片。否则，将会损坏数据。

接着，*/etc/vfstab* 文件中与该文件系统相对应的项将更改（如果是首次运行该命令，则输入该项），以便引用该串联。例如，*/etc/vfstab* 文件中最初显示下行：

```
/dev/dsk/c0t1d0s2 /dev/rdisk/c0t1d0s2 /docs ufs 2 yes -
```

该行应当更改为以下内容：

```
/dev/md/dsk/d25 /dev/md/rdsk/d25 /docs ufs 2 yes -
```

最后，将重新挂载该文件系统。

另请参见 对于 UFS 文件系统，请针对该串联运行 *growfs* 命令。请参见第 241 页中的“如何扩展文件系统”。

某些应用程序（如数据库）不使用文件系统，数据库之类的应用程序使用原始串联，且必须以自己的方式来识别串联或增加扩展空间。

▼ 如何扩展现有的 RAID-0 卷

使用串联条带化，可以扩展现有的条带。例如，如果某个条带的存储容量已经用完，则可以将其转换为串联的条带。这样可以扩展存储容量，且无需备份并恢复数据。

此过程假设要向现有的条带中再添加一个条带。



注意 – 如果您希望运行具有 32 位内核的 Solaris 软件，请不要创建大于 1TB 的卷。另外，如果您希望运行版本低于 Solaris 9 4/03 发行版的 Solaris OS，请不要创建大于 1TB 的卷。有关 Solaris Volume Manager 中多 TB 支持的更多信息，请参见第 45 页中的“Solaris Volume Manager 中的多 TB 支持的概述”。

开始之前 请查看第 45 页中的“创建 Solaris Volume Manager 组件的先决条件”和第 81 页中的“创建 RAID-0 卷的背景信息”。

D 要创建串联的条带，请使用以下方法之一：

- 从 Solaris Management Console 中增强的存储工具中，打开“卷”节点。选择“操作” ⇒ “创建卷”，然后按照向导中的说明操作。有关更多信息，请参见联机帮助。
- 要使用命令行串联现有的条带，请使用以下形式的 `metattach` 命令：

```
# metattach volume-name component-names
```

volume-name 指定要扩展的卷的名称。

component-names 指定所使用的组件的名称。如果使用多个组件，请用空格分隔它们。

有关更多信息，请参见 `metattach(1M)` 手册页。

示例 9-6 通过附加单个片来创建串联的条带

```
# metattach d2 c1t2d0s2
```

```
d2: components are attached
```

此示例说明如何向现有的条带 d2 附加片。系统会确认该片是否已附加。

示例 9-7 通过添加多个片来创建串联的条带

```
# metattach d25 c1t2d0s2 c1t2d1s2 c1t2d3s2
```

```
d25: components are attached
```

此示例提取现有的三向条带 d25 并将它与另一个三向条带串联起来。由于没有为附加的片提供交错值，因此这些条带将沿用为 d25 配置的交错值。系统会验证该卷是否已经设置。

另请参见 对于 UFS 文件系统，请针对卷运行 `growfs` 命令。请参见第 241 页中的“如何扩展文件系统”。

某些应用程序（如数据库）不使用文件系统，数据库之类的应用程序使用原始卷，且必须以自己的方式来识别卷或增加扩展空间。

要为文件系统准备新创建的串联条带，请参见《系统管理指南：设备和文件系统》中的第 18 章“创建 UFS、TMPFS 和 LOFS 文件系统（任务）”。

删除 RAID-0 卷

▼ 如何删除 RAID-0 卷

- 1 请确保您具有超级用户权限，而且拥有所有数据的最新备份。

- 2 请确定不再需要该卷。

如果删除某个条带或串联并重用已删除卷中的片，则该卷上的所有数据都将从系统中删除。

- 3 如有必要，请取消对文件系统的挂载。

```
# umount /filesystem
```

- 4 要删除卷，请使用以下方法之一：

- 从 Solaris Management Console 中增强的存储工具中，打开“卷”节点。选择“编辑” ⇒ “删除”，然后按照屏幕上的说明操作。有关更多信息，请参见联机帮助。
- 使用以下形式的 `metaclear` 命令删除卷：

```
metaclear volume-name
```

有关更多信息，请参见以下示例和 `metaclear(1M)` 手册页。

示例 9-8 删除串联

```
# umount d8
```

```
# metaclear d8
```

```
d8: Concat/Stripe is cleared
```

（编辑 `/etc/vfstab` 文件）

此示例说明如何删除包含一个已挂载的文件系统的串联 **d8**。必须先取消对该文件系统的挂载，才能删除该卷。系统会显示一条确认消息，声明该串联已删除。如果 `/etc/vfstab` 文件中存在与该卷相对应的项，请删除该项。如果卷已不存在，却要在该卷上挂载文件系统，会使系统出现混乱。

RAID-1（镜像）卷（概述）

本章说明与镜像和子镜像相关的基本 Solaris Volume Manager 概念。有关执行相关任务的信息，请参见第 11 章。

本章包含以下信息：

- 第 93 页中的 “RAID-1（镜像）卷概述”
- 第 96 页中的 “RAID-1 卷（镜像）重新同步”
- 第 97 页中的 “创建和维护 RAID-1 卷”
- 第 101 页中的 “引导到单用户模式对 RAID-1 卷的影响”
- 第 101 页中的 “方案—RAID-1 卷（镜像）”

RAID-1（镜像）卷概述

RAID-1 卷（或**镜像**）是用于维护 RAID-0（条带化或串联）卷中的相同数据副本的卷。各个镜像的 RAID-0 卷称为**子镜像**。镜像要求购买磁盘。所需的磁盘空间至少要为必须镜像的数据量的两倍。由于 Solaris Volume Manager 必须对所有子镜像写入数据，因此镜像还会增加将要写入的请求写入磁盘所需的时间量。

配置镜像后，就可以像使用物理片一样使用该镜像。

可以镜像任何文件系统，包括现有文件系统。这些文件系统包括根 (/)、swap 和 /usr。也可以将镜像用于任何应用程序，如数据库。

提示 – 将 Solaris Volume Manager 的热备用功能与镜像结合使用，可以使数据始终安全且可用。有关热备件的信息，请参见第 16 章和第 17 章。

子镜像概述

镜像由一个或多个称为子镜像的 RAID-0（条带化或串联）卷组成。

镜像最多可以包含四个子镜像。但是，双向镜像通常会为大多数应用程序提供足够的数据冗余，且就磁盘驱动器成本而言比较廉价。第三个子镜像使您可以在其中一个子镜像处于脱机状态下进行联机备份，同时又能保持数据冗余。

如果在“脱机”状态下获取子镜像，则镜像将停止向子镜像读取和写入数据。此时，可以访问子镜像本身（例如，执行备份）。但是，子镜像处于只读状态。子镜像处于脱机状态时，Solaris Volume Manager 将跟踪向该镜像中写入的所有内容。当使子镜像恢复联机状态时，只有在子镜像处于脱机状态时写入的镜像部分（**重新同步区域**）会被重新同步。还可以在脱机状态下获取子镜像，以对出现错误的物理设备进行故障排除或修复。

可以随时附加子镜像或从镜像拆离子镜像，但至少必须有一个子镜像始终处于连接状态。

通常，创建的镜像只包含一个子镜像。创建镜像后，可以附加第二个子镜像。

方案一 RAID-1（镜像）卷

图 10-1 说明了镜像 d20。该镜像由两个卷（子镜像）d21 和 d22 组成。

Solaris Volume Manager 在多个物理磁盘上生成相同的数据副本，并向应用程序显示一个虚拟磁盘（本示例中为 d20）。所有磁盘上写入的内容都是相同的。磁盘读取内容来自其中某个基础子镜像。镜像 d20 的总容量为最小子镜像的大小（如果子镜像的大小不相等）。

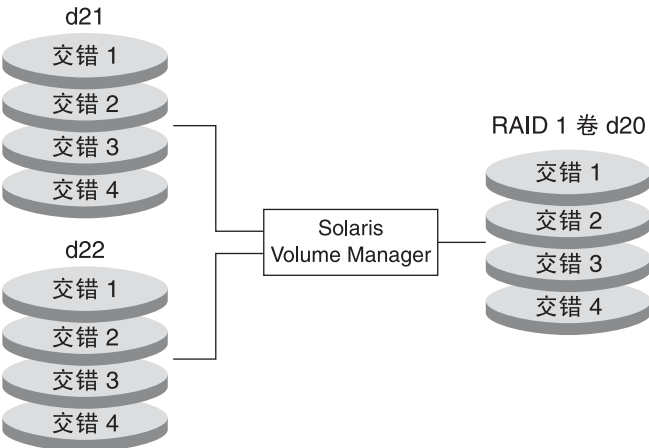


图 10-1 RAID-1（镜像）示例

提供 RAID-1+0 和 RAID-0+1

Solaris Volume Manager 支持 RAID-1+0 和 RAID-0+1 冗余。RAID-1+0 冗余代表先建立一组镜像配置，然后再进行条带化。RAID-0+1 冗余代表先建立一组条带配置，然后再进行镜像。Solaris Volume Manager 界面显示所有的 RAID-1 设备在严格意义上讲都为 RAID-0+1。但是，Solaris Volume Manager 可以识别出基础组件并对每个组件分别进行镜像（如果可能）。

注 - Solaris Volume Manager 无法始终提供 RAID-1+0 功能。但是，如果两个子镜像彼此相同且由磁盘片（而不是软分区）组成，则提供 RAID-1+0 功能是可能的。

请考虑使用由三个条带化片组成的双向镜像来实现 RAID-0+1。如果不使用 Solaris Volume Manager 产品，则一个片出现故障可能会导致镜像的一面失败。假设未使用任何热备件，则两个片出现故障将导致镜像失败。使用 Solaris Volume Manager 时，最多可有三个片出现故障，而不会导致镜像失败。由于这三个条带化片中的每个片均已单独镜像到另一半镜像的对应部分中，因此镜像不会失败。

图 10-2 说明了 RAID-1 卷遇到片丢失的危险但 RAID-1+0 功能阻止了数据丢失的这一过程。

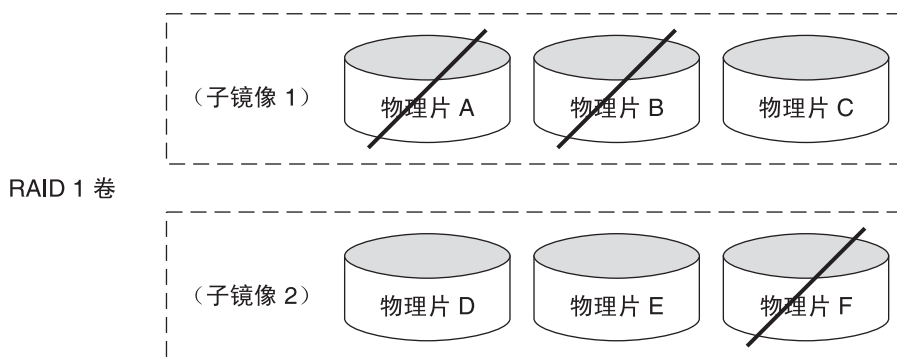


图 10-2 RAID-1+0 示例

RAID-1 卷包含两个子镜像。每个子镜像都包含三个相同的物理磁盘，且这些物理磁盘的交错值都相同。允许 A、B 和 F 三个磁盘出现故障。镜像的整个逻辑块范围仍然包含于至少一个良好的磁盘上。卷的所有数据都可用。

但是，如果磁盘 A 和 D 出现故障，则部分镜像数据在任何磁盘上都不再可用。对这些逻辑块的访问将失败。但是，对数据可用的镜像部分的访问仍将成功。在这种情况下，该镜像和出现了坏块的单个磁盘一样，损坏的部分不可用，但其余部分是可用的。

RAID-1 卷（镜像）重新同步

RAID-1 卷（镜像）重新同步是指在出现以下情况之一时将数据从一个子镜像复制到另一个子镜像的过程：

- 子镜像失败
- 系统崩溃
- 子镜像脱机之后又重新联机
- 添加了新的子镜像

执行重新同步时，用户始终可以读取和写入镜像。

镜像重新同步通过维护具有相同数据的所有子镜像（正在写入的镜像除外），确保了正确的镜像操作。

注 - 不应忽略镜像重新同步。不需要以手动方式启动镜像重新同步。此过程将自动执行。

完整重新同步

将新的子镜像附加（添加）到镜像时，该镜像中另一子镜像中的所有数据都将自动写入新附加的子镜像。完成镜像重新同步后，新的子镜像即可读取。拆离子镜像之前，该子镜像始终附加到镜像。

如果系统在重新同步期间崩溃了，则系统完成重新引导后就会重新启动重新同步。

优化的重新同步

出现系统故障后重新引导系统期间，或在子镜像脱机之后又重新联机时，Solaris Volume Manager 将执行**优化的镜像重新同步**。元盘驱动程序将跟踪子镜像区域。元盘驱动程序借助此功能来了解，哪些子镜像区域在出现故障后可能不同步。优化的镜像重新同步仅在不同步区域中执行。可以指定在重新引导期间重新同步镜像的顺序。通过将子镜像传送号设置为 0，可以省略镜像重新同步。有关与更改传送号关联的任务，请参见[示例 11-16](#)。



注意 - 应仅对以只读方式挂载的镜像使用传送号 0。

部分重新同步

替换子镜像内的片后，Solaris Volume Manager 将执行数据的**部分镜像重新同步**。Solaris Volume Manager 会将数据从另一子镜像的仍然完好的片复制到替换后的片中。

创建和维护 RAID-1 卷

本节提供可帮助您创建镜像的指南，此外，还提供所创建镜像的性能指南。

RAID-1 卷的配置指南

- 创建镜像之前，请先创建组成镜像的 RAID-0（条带化或串联）卷。
- 创建镜像时，请首先创建单向镜像，然后附加第二个子镜像。此策略可以启动重新同步操作，此外，还可以确保数据不会被损坏。您还可以创建单向镜像，以用作将来的双向或多向镜像。
- 可以使用单个命令根据单向镜像来创建双向镜像、三向镜像或四向镜像。通过使用单个命令来创建所有的子镜像，可以加速创建过程。只有在未镜像现有数据，而且已充分销毁所有子镜像上的数据时，才使用此过程。
- 可以使用在片上生成的现有文件系统来创建 RAID-1 卷。主 RAID-0 卷（子镜像）中只能包括单个片。如果您正在镜像根文件系统或其他对系统非常关键的文件系统，则所有的子镜像必须仅包含单个片。
- 使用 `swap -l` 命令可以检查所有的 `swap` 设备。指定为 `swap` 的每个片必须独立于其余交换片进行镜像。
- Solaris Management Console 中增强的存储工具 不支持取消镜像根 (`/`)、`/opt`、`/usr` 或 `swap`。实际上，该工具不支持取消镜像任何不能在系统处于运行状态时取消挂载的文件系统。请对这些文件系统改用命令行过程。
- 使用大小相同的子镜像。不同大小的子镜像会导致部分磁盘空间不能使用。
- 仅使用镜像内以类似方式配置的子镜像。需要特别指出的是，如果创建的镜像包含未加标记的子镜像，则不能附加包含磁盘标号的任何子镜像。
- 如果您的镜像文件系统中附加的第一个子镜像不是从柱面 0 开始的，则附加的所有其他子镜像也不得从柱面 0 开始。如果尝试附加从柱面 0 开始的子镜像，则会显示以下错误消息：

```
can't attach labeled submirror to an unlabeled mirror
```

要么在特定镜像内使用的所有子镜像都必须从柱面 0 开始，要么在特定镜像内使用的所有子镜像都不得从柱面 0 开始。

所有子镜像的开始柱面不必相同。但是，所有子镜像要么都包括柱面 0，要么都不包括柱面 0。

- 通过在创建镜像之前添加其他状态数据库副本，可以提高镜像的性能。按照一般规则，为添加到系统中的每个镜像添加两个其他副本。Solaris Volume Manager 使用这些附加副本存储脏区日志 (DRL, dirty region log)，该日志用于提供优化的重新同步。通过提供足够数量的副本，可以使 I/O 对 RAID-1 卷性能的影响降至最低程度。在相同的磁盘或控制器上使用至少两个副本作为副本记录的镜像，还可以帮助提高整体性能。

- 只需直接挂载镜像设备即可。请不要尝试直接挂载子镜像，除非该子镜像处于脱机状态且以只读方式挂载。请不要挂载作为子镜像一部分的片。此过程可能会销毁数据并使系统崩溃。

RAID-1 卷的性能指南

- 使不同子镜像的片位于不同的磁盘和控制器中。如果同一镜像的两个和多个子镜像的片位于同一个磁盘上，则对数据的保护作用将大大降低。同样，应将子镜像分布在不同的控制器中，因为控制器及关联电缆比磁盘更容易发生故障。此做法还可以提高镜像性能。
- 对一个镜像使用同一类型的磁盘和控制器。特别是对于旧的 SCSI 存储设备，不同型号或不同品牌的磁盘或控制器的性能差别很大。如果在单个镜像中使用具有不同性能级别的磁盘和控制器，则性能可能会明显降低。
- 镜像可能会提高读取性能，但写入性能始终会降低。仅在线程或异步 I/O 情况下，镜像才会提高读取性能。从卷单线程读取不会提高性能。
- 可以体验到使用镜像读取策略可提高性能。例如，缺省读取模式为以循环方式交替读取磁盘中的数据。此策略是缺省设置，因为循环往往最适合于 UFS 多用户、多处理器活动。

在某些情况下，`geometric` 读取选项通过使磁头移动和访问时间降至最低程度，可以提高性能。此选项在以下情况下最有效：

- 每个磁盘上只有一个片
- 一次只有一个进程使用片或文件系统
- I/O 模式高度连续或所有的访问权限都为读取权限
- 可以将子镜像附加到镜像上，而不中断服务。可以将子镜像附加到镜像上，以创建双向、三向和四向镜像。
- 使子镜像处于脱机状态时，可以防止镜像从子镜像读取数据并向子镜像写入数据。但是，要保留子镜像与镜像的逻辑关联。子镜像处于脱机状态时，Solaris Volume Manager 将跟踪向该镜像中写入的所有内容。子镜像恢复联机状态时，所有内容都将被写入该子镜像中。通过执行优化的重新同步，Solaris Volume Manager 只须重新同步已发生更改的数据，而不必重新同步整个子镜像。拆离子镜像时，需要提供该子镜像与对应镜像的逻辑关联。通常，要执行维护，需要使子镜像处于脱机状态。要删除子镜像，需要先拆离该子镜像。

关于 RAID-1 卷选项

以下选项可用于优化镜像性能：

- 镜像读取策略
- 镜像写入策略
- 重新同步镜像的顺序（传送号）

最初创建镜像时，可以定义镜像选项。设置镜像且镜像处于运行状态后，还可以更改镜像选项。有关与更改这些选项相关的任务，请参见第 136 页中的“[如何更改 RAID-1 卷选项](#)”。

RAID-1 卷读取和写入策略

Solaris Volume Manager 可以为 RAID-1 卷配置不同的读取和写入策略。正确设置读取和写入策略可以提高给定配置的性能。

表 10-1 RAID-1 卷读取策略

读取策略	说明
循环（缺省）	尝试平衡子镜像中的负载。从镜像的所有子镜像中依照循环顺序（一个接着一个）执行所有读取操作。
几何	可以基于逻辑磁盘块地址划分不同子镜像的读取操作。例如，使用双向子镜像时，可以将镜像上的磁盘空间划分为两个大小相等的逻辑地址范围。从某个子镜像的读取操作被限制到逻辑范围的一半。从其他子镜像的读取操作被限制到另一半。几何读取策略可以有效减少读取所需的查找时间。性能提高的程度（通过使用此读取策略）取决于系统 I/O 负载以及应用程序的访问模式。
第一	将所有读取操作导向第一个子镜像。只有在组成第一个子镜像的设备比第二个子镜像的设备速度明显更快时，才应使用此策略。

表 10-2 RAID-1 卷写入策略

写入策略	说明
并行（缺省）	对镜像执行同时复制并分发到所有子镜像的写入操作。
串行	对子镜像连续执行写入操作（也就是说，第一个子镜像写入操作完成后，才能启动第二个子镜像写入操作）。此策略指定必须完成向一个子镜像的写入操作，才能启动下一个子镜像写入操作。如果因电源故障等无法访问子镜像，则系统会提供此策略。

传送号

传送号（范围为 0-9 的编号）可以确定在系统重新引导期间重新同步特定镜像的顺序。缺省传送号为 1。首先重新同步较低的传送号。如果使用 0，则将跳过镜像重新同步。应仅对以只读方式挂载的镜像使用传送号 0。具有相同传送号的镜像同时重新同步。

了解子镜像状态以确定维护操作

Solaris Volume Manager 的 `metastat` 命令可用于报告有关 RAID 1 卷和子镜像的状态信息。状态信息可帮助您确定是否需要 RAID-1 卷执行维护操作。下表说明了 RAID-1 卷运行 `metastat` 命令时显示的子镜像状态。

表 10-3 子镜像状态

状态	含义
正常	子镜像没有错误且正常发挥作用。
正在重新同步	正在以活动方式重新同步子镜像。出现错误且已更正错误，子镜像刚刚恢复联机状态，或者添加了新的子镜像。
需要维护	子镜像中的片遇到 I/O 错误或打开错误。子镜像中的此片的所有读取和写入操作都已停止。

此外，对于子镜像中的每一片，`metastat` 命令都会显示以下信息：

- 设备 指示条带中的片的设备名称
- 起始块 指示片开始的块
- Dbase 指示片中是否包含状态数据库副本
- 状态 指示片的状态
- 热备件 指示正在将片用作出现故障片的热备件

子镜像状态仅提供有关子镜像状态的常规信息。排除镜像错误时，片状态可能是要查阅的最重要的信息。如果子镜像报告“需要维护”状态，则必须参阅片状态来了解更多信息。

根据片处于“维护”状态还是处于“最近出错”状态，可以执行不同的恢复操作。如果只有处于“维护”状态的片，则可以按任意顺序修复这些片。如果同时具有处于“维护”状态和处于“最近出错”状态的片，则必须首先修复处于“维护”状态的片。修复处于“维护”状态的片后，请接着修复处于“最近出错”状态的片。有关更多信息，请参见第 242 页中的“在 RAID-1 和 RAID-5 卷中替换和启用组件的概述”。

下表说明了子镜像的片状态以及可能执行的操作。

表 10-4 子镜像片状态

状态	含义	操作
正常	片没有错误且正常发挥作用。	无。
正在重新同步	正在以活动方式重新同步片。出现错误且已更正错误，子镜像刚刚恢复联机状态，或者添加了新的子镜像。	如果需要，请监视子镜像状态，直到重新同步完成为止。

表 10-4 子镜像片状态 （续）

状态	含义	操作
维护	片遇到 I/O 错误或打开错误。此组件的所有读取和写入操作都已停止。	启用或替换出现故障的片。请参见第 132 页中的“如何启用子镜像中的片”或第 138 页中的“如何替换子镜像中的片”。 <code>metastat</code> 命令将显示 <code>invoke</code> 恢复消息，其中包含要使用 <code>metareplace</code> 命令执行的相应操作。也可以使用 <code>metareplace -e</code> 命令。
最近出错	片遇到 I/O 错误或打开错误。但是，由于其他片出现了故障，因此不会在其他位置复制数据。仍然对片执行 I/O 操作。如果产生了 I/O 错误，则镜像 I/O 将失败。	首先，启用或更改处于“维护”状态的片。请参见第 132 页中的“如何启用子镜像中的片”或第 138 页中的“如何替换子镜像中的片”。通常，此错误会导致某些数据丢失，因此修复镜像之后，应验证该镜像。对于文件系统，请使用 <code>fsck</code> 命令，然后检查数据。应用程序或数据库必须具有各自的设备验证方法。

引导到单用户模式对 RAID-1 卷的影响

有时，您可能需要将包含根 (/)、/usr 和 swap（即所谓的“引导”文件系统）镜像的系统引导到单用户模式（通过使用 `boot -s` 命令）。在这种情况下，这些镜像（可能还有系统上的所有镜像）都会以“需要维护”状态出现（如果使用 `metastat` 命令进行查看）。此外，如果对这些片执行写入操作，则 `metastat` 命令将显示镜像中的脏区数目不断增加。

这种情况看起来存在潜在危险。但是，当将系统引导到单用户模式时，`metasync -r` 命令（通常在引导系统以重新同步镜像时运行）将被中止。重新引导系统后，`metasync -r` 命令将运行并重新同步所有镜像。

如果您担心此情况，则可以手动运行 `metasync -r` 命令。

方案－RAID-1 卷（镜像）

RAID-1 卷提供了一种构造冗余卷的方式。因此，当某个基础 RAID-0 卷部分或完全出现故障时，不会丢失数据或中断对文件系统的访问。以下示例（利用在第 5 章中说明且在第 81 页中的“方案－RAID-0 卷”中继续使用的方案）介绍了 RAID-1 卷如何提供冗余存储。

正如第 81 页中的“方案－RAID-0 卷”中所述，样例系统具有两个 RAID-0 卷。每个卷的大小大约为 27 GB，且跨越三个磁盘。通过创建 RAID-1 卷来镜像这两个 RAID-0 卷，完全冗余存储空间可以提供有弹性的数据存储。

在该 RAID-1 卷中，磁盘控制器的故障不会中断对卷的访问。同样，在不中断访问的情况下，最多可能允许三个单独的磁盘出现故障。

要针对可能中断访问的问题提供进一步的保护，请使用热备件，如第 16 章中所述。有关具体情况，请参见第 180 页中的“热备件的工作原理”。

RAID-1（镜像）卷（任务）

本章介绍如何执行与 RAID-1 卷相关的 Solaris Volume Manager 任务。有关相关概念的信息，请参见第 10 章。

RAID-1 卷（任务列表）

以下任务列表列出了管理 Solaris Volume Manager RAID-1 卷所需的过程。

任务	说明	参考
从未使用的片创建镜像	使用 Solaris Volume Manager GUI 或 <code>metainit</code> 命令从未使用的片创建镜像。	第 104 页中的“如何从未使用的片创建 RAID-1 卷”
从现有文件系统创建镜像	使用 Solaris Volume Manager GUI 或 <code>metainit</code> 命令从现有文件系统创建镜像。	第 107 页中的“如何从文件系统创建 RAID-1 卷”
从根 (/) 文件系统创建镜像	使用 Solaris Volume Manager GUI 或 <code>metainit</code> 命令从根 (/) 文件系统创建镜像。	第 112 页中的“SPARC: 如何从根 (/) 文件系统创建 RAID-1 卷” 第 121 页中的“x86: 如何使用 DCA 从根 (/) 文件系统创建 RAID-1 卷”
附加子镜像	使用 Solaris Volume Manager GUI 或 <code>metattach</code> 命令附加子镜像。	第 128 页中的“如何附加子镜像”
拆离子镜像	使用 Solaris Volume Manager GUI 或 <code>metadetach</code> 命令拆离子镜像。	第 130 页中的“如何拆离子镜像”

任务	说明	参考
使子镜像联机或脱机	使用 Solaris Volume Manager GUI 或 <code>metaonline</code> 命令将子镜像联机。使用 Solaris Volume Manager GUI 或 <code>metaoffline</code> 命令使子镜像脱机。	第 130 页中的 “如何使子镜像脱机和联机”
启用子镜像内的片	使用 Solaris Volume Manager GUI 或 <code>metareplace</code> 命令启用子镜像内的片。	第 132 页中的 “如何启用子镜像中的片”
检查镜像状态	使用 Solaris Volume Manager GUI 或 <code>metastat</code> 命令检查 RAID-1 卷的状态。	第 132 页中的 “如何查看镜像和子镜像的状态”
更改镜像选项	使用 Solaris Volume Manager GUI 或 <code>metaparam</code> 命令更改特定 RAID-1 卷的选项。	第 136 页中的 “如何更改 RAID-1 卷选项”
扩展镜像	使用 Solaris Volume Manager GUI 或 <code>metattach</code> 命令扩展镜像的容量。	第 137 页中的 “如何扩展 RAID-1 卷”
替换子镜像内的片	使用 Solaris Volume Manager GUI 或 <code>metareplace</code> 命令替换子镜像内的片。	第 138 页中的 “如何替换子镜像中的片”
替换子镜像	使用 Solaris Volume Manager GUI 或 <code>metattach</code> 命令替换子镜像。	第 140 页中的 “如何替换子镜像”
删除镜像（取消镜像）	使用 Solaris Volume Manager GUI、 <code>metadetach</code> 命令或 <code>metaclear</code> 命令取消对文件系统的镜像。	第 142 页中的 “如何取消镜像文件系统”
删除无法取消挂载的文件系统的镜像（取消镜像）	使用 Solaris Volume Manager GUI、 <code>metadetach</code> 命令或 <code>metaclear</code> 命令取消对无法取消挂载的文件系统的镜像。	第 145 页中的 “如何取消镜像无法挂载的文件系统”
使用镜像执行备份	使用 Solaris Volume Manager GUI、 <code>metaonline</code> 命令或 <code>metaoffline</code> 命令通过镜像执行备份。	第 149 页中的 “如何对 RAID-1 卷执行联机备份”

创建 RAID-1 卷

▼ 如何从未使用的片创建 RAID-1 卷

以下过程说明如何创建双向镜像。该过程也可用于创建三向或四向镜像。

开始之前 请查看第 45 页中的 “创建 Solaris Volume Manager 组件的先决条件” 和第 97 页中的 “创建和维护 RAID-1 卷”。

1 创建两个条带化或串联的卷，这些组件会成为子镜像。

请参见第 84 页中的“如何创建 RAID-0（条带化）卷”或第 85 页中的“如何创建 RAID-0（串联）卷”。

2 要创建镜像，请使用以下方法之一：

- 从 Solaris Management Console 中增强的存储工具中，打开“卷”节点，然后选择“操作”⇒“创建卷”并按照屏幕上的说明操作。有关更多信息，请参见联机帮助。
- 使用以下格式的 `metainit` 命令创建单向镜像：

```
# metainit volume-name -m submirror-name
volume-name      指定要创建的卷的名称
-m              指定要创建镜像
submirror-name   指定将成为镜像中的第一个子镜像的组件的名称
```

有关更多信息，请参见以下示例和 `metainit(1M)` 手册页。

3 要添加第二个子镜像，请使用以下方法之一：

- 从 Solaris Management Console 中增强的存储工具中，打开“卷”节点，然后选择要修改的镜像。选择“操作”⇒“属性”，然后选择“子镜像”。按照屏幕上的说明附加子镜像。有关更多信息，请参见联机帮助。
- 使用以下格式的 `metattach` 命令：

```
# metattach volume-name submirror-name
volume-name      指定要向其中添加子镜像的 RAID-1 卷的名称
submirror-name   指定将成为附加到该镜像的第二个子镜像的组件的名称
```

有关更多信息，请参见以下示例和 `metattach(1M)` 手册页。

示例 11-1 创建双向镜像

```
# metainit d51 1 1 c0t0d0s2
```

```
d51: Concat/Stripe is setup
```

```
# metainit d52 1 1 c1t0d0s2
```

```
d52: Concat/Stripe is setup
```

```
# metainit d50 -m d51
```

```
d50: Mirror is setup
```

```
# metattach d50 d52
```

```
d50: Submirror d52 is attached
```

以下示例说明如何创建双向镜像 **d50**。可以使用 **metainit** 命令创建两个子镜像（**d51** 和 **d52**），即 RAID-0 卷。可以使用 **metainit -m** 命令从 **d51** RAID-0 卷创建单向镜像。可以使用 **metattach** 命令附加 **d52**，从而创建一个双向镜像并导致重新同步。在重新同步的过程中，所附加的子镜像中的任何数据都会被其他子镜像覆写。

示例 11-2 创建无需重新同步的双向镜像

```
# metainit d51 1 1 c0t0d0s2
```

```
d51: Concat/Stripe is setup
```

```
# metainit d52 1 1 c1t0d0s2
```

```
d52: Concat/Stripe is setup
```

```
# metainit d50 -m d51 d52
```

```
metainit: d50: WARNING: This form of metainit is not recommended.
```

```
The submirrors may not have the same data.
```

```
Please see ERRORS in metainit(1M) for additional information.
```

```
d50: Mirror is setup
```

本示例说明如何创建双向镜像 **d50**。可以使用 **metainit** 命令创建两个子镜像（**d51** 和 **d52**），即 RAID-0 卷。然后，针对这两个子镜像运行 **metainit -m** 命令以创建镜像。如果使用 **metainit** 命令而非 **metattach** 命令创建镜像，则不会执行重新同步操作。因此，当 Solaris Volume Manager 假定镜像的两面相同并且可以互换使用时，数据可能会损坏。

另请参见 要为文件系统准备新创建的镜像，请参见《系统管理指南：设备和文件系统》中的第 18 章“创建 UFS、TMPFS 和 LOFS 文件系统（任务）”。某些应用程序（如数据库）不使用文件系统，而是使用原始设备。这些应用程序必须能够以自己的方式来访问原始设备。

▼ 如何从文件系统创建 RAID-1 卷

使用本过程可以对现有文件系统进行镜像。如果该文件系统可以取消挂载，则无需重新引导即可完成整个过程。对于无法取消挂载的文件系统（如 `/usr` 和 `/swap`），必须重新引导系统才能完成该过程。

如果从构建于磁盘片上的现有文件系统创建 RAID-1 卷，则主 RAID-0 卷（子镜像）中只能包含这一个片。如果要镜像对系统非常关键的文件系统，则所有的子镜像都必须仅包含一个片。

有关与对根 (`/`) 文件系统进行镜像相关联的过程，请参见第 112 页中的“SPARC: 如何从根 (`/`) 文件系统创建 RAID-1 卷”和第 121 页中的“x86: 如何使用 DCA 从根 (`/`) 文件系统创建 RAID-1 卷”。

在本过程使用的示例中，现有的片是 `c1t0d0s0`。第二个片 `c1t1d0s0` 适用于镜像的另一半。子镜像分别是 `d1` 和 `d2`，镜像是 `d0`。



注意 – 请确保使用 `metainit` 命令创建单向镜像，然后使用 `metattach` 命令附加其他子镜像。如果不使用 `metattach` 命令，则不会执行重新同步操作。因此，当 Solaris Volume Manager 假定镜像的两面相同并且可以互换使用时，数据可能会损坏。

开始之前

请查看第 45 页中的“创建 Solaris Volume Manager 组件的先决条件”和第 97 页中的“创建和维护 RAID-1 卷”。

- 1 确定包含要进行镜像的现有文件系统的片，本示例使用片 `c1t0d0s0`。
- 2 使用以下方法之一，在上一步中的片上创建一个新的 RAID-0 卷：
 - 从 Solaris Management Console 中增强的存储工具中，打开“卷”节点，然后选择“操作”⇒“创建卷”。请按照屏幕上的说明操作。有关更多信息，请参见联机帮助。
 - 使用以下格式的 `metainit` 命令：

```
# metainit -f volume-name number-of-stripes components-per-stripe component-name
```

`-f` 强制继续执行该命令。如果片中包含已挂载的文件系统，则必须使用此选项。

`volume-name` 指定要创建的卷的名称。有关对卷进行命名的信息，请参见第 42 页中的“卷名称”。

`number-of-stripes` 指定要创建的条带的数目。

`components-per-stripe` 指定每个条带应包含的组件的数目。

`component-names` 指定所使用的组件的名称。本示例使用根片 `c0t0d0s0`。

- 3 在未使用的片（在本示例中为 `clt1d0s0`）上创建另一个 RAID-0 卷（串联），将其用作第二个子镜像。第二个子镜像不得小于原来的子镜像的大小。请使用以下方法之一：

- 从 Solaris Management Console 中增强的存储工具中，打开“卷”节点，然后选择“操作”->“创建卷”并按照屏幕上的说明操作。有关更多信息，请参见联机帮助。
- 使用以下格式的 `metainit` 命令：

```
# metainit volume-name number-of-stripes components-per-stripe component-name
```

注 - 有关这些选项的说明，请参见步骤 2。

- 4 使用以下方法之一创建单向镜像：

- 从 Solaris Management Console 中增强的存储工具中，打开“卷”节点，然后选择“操作”⇒“创建卷”。请按照屏幕上的说明操作。有关更多信息，请参见联机帮助。
- 使用以下格式的 `metainit` 命令：

```
# metainit volume-name -m submirror-name
```

`volume-name` 指定要创建的卷的名称。

`-m` 指定要创建镜像。

`submirror-name` 指定将成为镜像中的第一个子镜像的组件的名称。在本示例中，这是包含根片的 RAID-0 卷。

有关更多信息，请参见 `metainit(1M)` 手册页。



注意 - 从现有文件系统创建镜像时，必须严格按照以下两个步骤操作以避免数据损坏。

- 5 编辑 `/etc/vfstab` 文件，以便文件系统挂载指令引用镜像，而不引用块设备。有关 `/etc/vfstab` 文件的更多信息，请参见《系统管理指南：设备和文件系统》中的“挂载文件系统”。

例如，如果 `/etc/vfstab` 文件包含文件系统的以下项：

```
/dev/dsk/slice /dev/rdisk/slice /var ufs 2 yes -
```

请更改该项，使其显示为：

```
/dev/md/dsk/mirror-name /dev/md/rdisk/mirror-name /var ufs 2 yes -
```

- 6 按照以下过程之一重新挂载新镜像的文件系统：

- 如果要对可以取消挂载的文件系统进行镜像，则取消挂载该文件系统并将其重新挂载。

```
# umount /filesystem
```

```
# mount /filesystem
```

- 如果要对无法取消挂载的文件系统进行镜像，则重新引导系统。

```
# reboot
```

7 使用以下格式的 `metattach` 命令附加第二个子镜像。

```
# metattach volume-name submirror-name
```

volume-name 指定要向其中添加子镜像的 RAID-1 卷的名称

submirror-name 指定将成为附加到该镜像的第二个子镜像的组件的名称

有关更多信息，请参见 `metattach(1M)` 手册页。

示例 11-3 从可以取消挂载的文件系统创建双向镜像

```
# metainit -f d1 1 1 c1t0d0s0
```

```
d1: Concat/Stripe is setup
```

```
# metainit d2 1 1 c1t1d0s0
```

```
d2: Concat/Stripe is setup
```

```
# metainit d0 -m d1
```

```
d0: Mirror is setup
```

```
# umount /master
```

(编辑 `/etc/vfstab` 文件使该文件系统引用镜像)

```
# mount /master
```

```
# metattach d0 d2
```

```
d0: Submirror d2 is attached
```

在本示例中，`-f` 选项会强制创建第一个串联 `d1`，其中包含已在 `/dev/dsk/c1t0d0s0` 上挂载的文件系统 `/master`。第二个串联 `d2` 是从 `/dev/dsk/c1t1d0s0` 创建的。该片不得小于 `d1` 的大小。带有 `-m` 选项的 `metainit` 命令可用来从 `d1` 创建单向镜像 `d0`。

接下来，应当在 `/etc/vfstab` 文件中更改与文件系统对应的项，使其引用该镜像。
`/etc/vfstab` 文件中以下行最初如下所示：

```
/dev/dsk/c1t0d0s0 /dev/rdsk/c1t0d0s0 /var ufs 2 yes -
```

该项更改为如下所示：

```
/dev/md/dsk/d0 /dev/md/rdsk/d0 /var ufs 2 yes -
```

最后，重新挂载该文件系统，并将子镜像 d2 附加到该镜像，这会导致重新同步镜像。系统会确认是否已设置 RAID-0 和 RAID-1 卷并已附加子镜像 d2。

示例 11-4 从无法取消挂载的文件系统创建双向镜像

```
# metainit -f d12 1 1 c0t3d0s6
```

```
d12: Concat/Stripe is setup
```

```
# metainit d22 1 1 c1t0d0s6
```

```
d22: Concat/Stripe is setup
```

```
# metainit d2 -m d12
```

```
d2: Mirror is setup
```

（编辑 `/etc/vfstab` 文件使 `/usr` 引用该镜像）

```
# reboot
```

```
...
```

```
# metattach d2 d22
```

```
d2: Submirror d22 is attached
```

本示例使用包含 `/usr` 文件系统的片创建双向镜像。`-f` 选项会强制创建第一个串联 d12，其中包含已在 `/dev/dsk/c0t3d0s6` 上挂载的文件系统 `/usr`。第二个串联 d22 是从 `/dev/dsk/c1t0d0s6` 创建的。该片不得小于 d12 的大小。带有 `-m` 选项的 `metainit` 命令使用包含 `/usr` 文件系统的串联创建单向镜像 d2。接下来，必须编辑 `/etc/vfstab` 文件以更改 `/usr` 的项，使其引用该镜像。

`/etc/vfstab` file 包含 `/usr` 文件系统的以下项：

```
/dev/dsk/c0t3d0s6 /dev/rdsk/c0t3d0s6 /usr ufs 1 yes -
```

将该项更改为如下所示：

```
/dev/md/dsk/d2 /dev/md/rdsk/d2 /usr ufs 1 yes -
```

重新引导之后，第二个子镜像 d22 将附加到该镜像，这会导致对镜像重新同步。

示例 11-5 从 /swap 空间创建镜像

```
# metainit -f d11 1 1 c0t0d0s1
```

```
d11: Concat/Stripe is setup
```

```
# metainit d21 1 1 c1t0d0s1
```

```
d21: Concat/Stripe is setup
```

```
# metainit d1 -m d11
```

```
d1: Mirror is setup
```

(编辑 `/etc/vfstab` 文件使 `swap` 引用该镜像)

```
# reboot
```

```
...
```

```
# metattach d1 d21
```

```
d1: Submirror d21 is attached
```

在本示例中，`-f` 选项会强制创建第一个串联 `d11`，其中包含已在 `/dev/dsk/c0t0d0s1` 上挂载的文件系统 `swap`。第二个串联 `d21` 是从 `/dev/dsk/c1t0d0s1` 创建的。该片不得小于 `d11` 的大小。带有 `-m` 选项的 `metainit` 命令可使用包含 `swap` 的串联创建单向镜像 `d1`。接下来，如果 `/etc/vfstab` 文件中存在与 `swap` 对应的项，则必须编辑该项，使其引用该镜像。

`/etc/vfstab` file 包含 `swap` 空间的以下项：

```
/dev/dsk/c0t0d0s1 - - swap - no -
```

请更改该项，使其显示为：

```
/dev/md/dsk/d1 - - swap - no -
```

重新引导之后，第二个子镜像 `d21` 将附加到该镜像，这会导致对镜像重新同步。

要在对 `swap` 空间进行镜像之后保存崩溃转储，请使用 `dumpadm` 命令将转储设备配置为卷。例如，如果交换设备的名称为 `/dev/md/dsk/d2`，请使用 `dumpadm` 命令将该设备设置为转储设备。

▼ SPARC: 如何从根 (/) 文件系统创建 RAID-1 卷

对 SPARC 平台上的根 (/) 文件系统进行镜像的过程与对无法取消挂载的其他任何文件系统进行镜像的过程相似，区别在于前者运行 `metaroot` 命令，而不是手动编辑 `/etc/vfstab` 文件。对根 (/) 文件系统进行镜像还需要记录备用引导设备的路径。如果子镜像失败，此设备会重新引导系统。

在本过程使用的示例中，现有的片是 `c1t0d0s0`。第二个片 `c1t1d0s0` 适用于镜像的另一半。子镜像分别是 `d1` 和 `d2`，镜像是 `d0`。



注意 – 请确保使用 `metainit` 命令创建单向镜像，然后使用 `metattach` 命令附加其他子镜像。如果不使用 `metattach` 命令，则不会执行重新同步操作。因此，当 Solaris Volume Manager 假定镜像的两面相同并且可以互换使用时，数据可能会损坏。

开始之前 请查看第 45 页中的“创建 Solaris Volume Manager 组件的先决条件”和第 97 页中的“创建和维护 RAID-1 卷”。

- 1 确定包含要进行镜像的现有根 (/) 文件系统的片。本示例使用片 `c1t0d0s0`。
- 2 使用以下方法之一，在上一步中的片上创建一个新的 RAID-0 卷。RAID-0 卷中只能包含一个片。
 - 从 Solaris Management Console 中增强的存储工具中，打开“卷”节点，然后选择“操作” ⇒ “创建卷”。请按照屏幕上的说明操作。有关更多信息，请参见联机帮助。
 - 使用以下格式的 `metainit` 命令：

```
# metainit -f volume-name number-of-stripes components-per-stripe component-name
```

<code>-f</code>	强制继续执行该命令。如果片中包含已挂载的文件系统，则必须使用此选项。
<code>volume-name</code>	指定要创建的卷的名称。有关对卷进行命名的信息，请参见第 42 页中的“卷名称”。
<code>number-of-stripes</code>	指定要创建的条带的数目。
<code>components-per-stripe</code>	指定每个条带应包含的组件的数目。
<code>component-names</code>	指定所使用的组件的名称。本示例使用根片 <code>c0t0d0s0</code> 。

- 3 在未使用的片（在本示例中为 `c1t1d0s0`）上创建另一个 RAID-0 卷，将其用作第二个子镜像。辅助子镜像不得小于原来的子镜像的大小。请使用以下方法之一：
 - 从 Solaris Management Console 中增强的存储工具中，打开“卷”节点，然后选择“操作” -> “创建卷”并按照屏幕上的说明操作。有关更多信息，请参见联机帮助。

- 使用以下格式的 `metainit` 命令：

```
# metainit volume-name number-of-stripes components-per-stripe component-name
```

注 – 有关这些选项的说明，请参见步骤 2。

4 使用以下方法之一创建单向镜像：

- 从 Solaris Management Console 中增强的存储工具中，打开“卷”节点，然后选择“操作”⇒“创建卷”。请按照屏幕上的说明操作。有关更多信息，请参见联机帮助。
- 使用以下格式的 `metainit` 命令：

```
# metainit volume-name -m submirror-name
```

volume-name 指定要创建的卷的名称。

-m 指定要创建镜像。

submirror-name 指定将成为镜像中的第一个子镜像的组件的名称。在本示例中，这是包含根片的 RAID-0 卷。

5 重新挂载刚进行镜像的文件系统。运行 `metaroot volume-name` 命令，将 *volume-name* 替换为已创建的镜像的名称。然后重新引导系统。

```
# metaroot volume-name
```

```
# reboot
```

有关更多信息，请参见 `metaroot(1M)` 手册页。

6 使用以下格式的 `metattach` 命令附加第二个子镜像。

```
# metattach volume-name submirror-name
```

volume-name 指定要向其中添加子镜像的 RAID-1 卷的名称

submirror-name 指定将成为附加到该镜像的第二个子镜像的组件的名称

有关更多信息，请参见 `metattach(1M)` 手册页。

7 记录替代引导路径。

- a. 确定备用根设备的路径。针对附加作为根 (/) 文件系统镜像的第二个子镜像的片，使用 `ls -l` 命令。

```
# ls -l /dev/dsk/ct1td0s0
```

```
lrwxrwxrwx 1 root root 55 Mar 5 12:54 /dev/rdisk/ct1td0s0 -> \
```

```
../../devices/sbus@1,f8000000/esp@1,200000/sd@3,0:a
```

- b. 记录 `/devices` 目录后面的字符串：`/sbus@1,f8000000/esp@1,200000/sd@3,0:a`。

注-由于系统可能不可用，因此还应当将这些信息记录在系统以外的某个位置。有关从备用引导设备引导的详细信息，请参见第 299 页中的“从引导问题中恢复”。

- c. 编辑该字符串，将主名称（在本示例中为 `sd`）更改为 `disk`，产生的结果为 `/sbus@1,f8000000/esp@1,200000/disk@3,0:a`。如果系统使用 IDE 总线，则最初的全路径可能如下所示：

```
$ ls -l /dev/dsk/c1t1d0s0
```

```
lrwxrwxrwx 1 root root 38 Mar 13 15:03 /dev/dsk/c0t0d0s0 -> \
```

```
../../devices/pci@1f,0/ide@d/dad@0,0:a
```

将主名称 `dad` 更改为 `disk` 之后，结果将为 `/pci@1f,0/ide@d/disk@0,0:a`

- d. 使用 **OpenBoot™ PROM** `nvalias` 命令为辅助根 (/) 文件系统镜像定义“备份根”设备别名。例如：

```
ok nvalias backup_root /sbus@1,f8000000/esp@1,200000/disk@3,0:a
```

- e. 重新定义 `boot-device` 别名，使其按照主子镜像和辅助子镜像的使用顺序来进行引用，并存储该配置。

```
ok printenv boot-device
```

```
boot-device =          disk net
```

```
ok setenv boot-device disk backup_root net
```

```
boot-device =          disk backup_root net
```

```
ok nvstore
```

注-如果主子镜像失败，系统将自动引导到第二个子镜像。或者，如果手动引导而不是使用自动引导，则可输入以下内容：

```
ok boot backup_root
```

示例 11-6 SPARC: 从根 (/) 文件系统创建镜像

```
# metainit -f d1 1 1 c0t0d0s0

d1: Concat/Stripe is setup

# metainit d2 1 1 c0t1d0s0

d2: Concat/Stripe is setup

# metainit d0 -m d1

d0: Mirror is setup

# metaroot d0

# lockfs -fa

# reboot

...

# metattach d0 d2

d0: Submirror d2 is attached

# ls -l /dev/dsk/c0t1d0s0

lrwxrwxrwx  1 root    root          88 Feb  8 15:51 /dev/dsk/c1t3d0s0 ->
../../devices/pci@1f,0/pci@1,1/ide@3/dad@0,0:a

# init 0

.

.

.

ok nvalias backup_root /pci@1f,0/pci@1,1/ide@3/disk@0,0:a

ok setenv boot-device disk backup_root net

ok nvstore
```

在本示例中，`-f` 选项会强制创建第一个 RAID-0 卷 `d1`，其中包含已在 `/dev/dsk/c0t0d0s0` 上挂载的根 (`/`) 文件系统。第二个串联 `d2` 是从 `/dev/dsk/c0t1d0s0` 创建的。该片不得小于 `d1` 的大小。带有 `-m` 选项的 `metainit` 命令可使用包含根 (`/`) 的串联创建单向镜像 `d0`。

接下来，使用 `metaroot` 命令编辑 `/etc/vfstab` 和 `/etc/system` 文件，以便可以使用卷上的根 (`/`) 文件系统引导系统。重新引导之前最好先运行 `lockfs -fa` 命令。有关更多信息，请参见 `lockfs(1M)` 手册页。

请勿在重新引导系统之前附加第二个子镜像。必须在运行 `metaroot` 命令之后和附加第二个子镜像之前重新引导系统。

重新引导之后，子镜像 `d2` 将附加到该镜像，这会导致对镜像重新同步。系统会确认是否已设置串联和镜像以及是否已附加子镜像 `d2`。

可以针对根原始设备运行 `ls -l` 命令，以确定备用根设备的路径，以备系统以后通过该路径进行引导。

x86: 从根 (`/`) 文件系统创建 RAID-1 卷

从 Solaris 10 1/06 发行版开始，GRand Unified Bootloader (GRUB) 已经取代了基于 x86 的系统中用于引导过程和配置的 Device Configuration Assistant (DCA)。有关此功能及其引入的增强功能的简要说明，请参阅《Solaris 10 新增功能》中的“基于 GRUB 的引导”。

本节中的过程介绍了从根 (`/`) 文件系统创建 RAID-1 卷的步骤。如果系统运行的是 Solaris 10 1/06 OS 或后续发行版，请按照第一个过程（使用 GRUB）中的步骤操作。否则，请执行第二个过程（使用 DCA）中的步骤。

在基于 x86 的系统上对根 (`/`) 文件系统进行镜像的过程与在 SPARC 系统上对根 ([code]) 文件系统进行镜像的过程相似。但是，在基于 x86 的系统上，BIOS 和 `fdisk` 分区会使镜像过程更复杂。

在这些过程使用的示例中，现有的片是 `c1t0d0s0`。第二个片 `c1t1d0s0` 适用于镜像的另一半。子镜像分别是 `d1` 和 `d2`，镜像是 `d0`。

注 - 在实现任何过程之前，请查看第 45 页中的“创建 Solaris Volume Manager 组件的先决条件”和第 97 页中的“创建和维护 RAID-1 卷”。

▼ x86: 如何使用 GRUB 从根 (/) 文件系统创建 RAID-1 卷

1 验证是否可将 BIOS 引导设备的顺序配置为允许系统从镜像中的第二块磁盘引导。

启动内核之前，系统由只读存储器 (read-only-memory, ROM) 基本输入/输出系统 (Basic Input/Output System, BIOS) 控制，该系统是基于 x86 的系统上的固件接口。BIOS 与基于 SPARC 的系统上的引导 PROM 类似。以下是一些 BIOS 任务：

- 执行启动功能。
- 检测可从中引导系统的正确设备。
- 从该设备装入主引导记录，以允许系统自行引导。

通常可以将 BIOS 配置为选择要探测其引导记录的设备的顺序。此外，现在实现的大多数 BIOS 都允许将设备配置为自动向辅助子镜像进行故障转移。如果系统的 BIOS 没有此功能，则主子镜像失败时，需要在系统引导过程中访问 BIOS，以便将系统重新配置为从辅助根片引导。有关如何配置 BIOS 中的设置的说明，请查阅 BIOS 的用户指南。

设置根镜像之前，请检查系统上的 BIOS，验证是否可以从多块磁盘引导。某些设备驱动程序配置为只能看到系统上的一个磁盘。

2 验证 fdisk 分区是否配置为支持根镜像。

如果对根 (/) 文件系统进行镜像时存在单独的 x86 引导分区，则会出现问题。因为 x86 引导分区位于 Solaris fdisk 分区之外，所以不能通过 Solaris Volume Manager 对 x86 引导分区进行镜像。此外，由于仅存在 x86 引导分区的一个副本，因此它还表示一个单点故障。

Solaris 10 1/06 软件和后续发行版的基于 GRUB 的安装程序不会再自动创建 x86 引导分区。但是，如果系统中已经存在 x86 引导分区，则安装程序在缺省情况下会保留该分区。

要确定系统是否有单独的 x86 引导分区，请检查 /etc/vfstab 文件。如果该文件中包含类似如下的项，则表明存在 x86 引导分区：

```
/dev/dsk/c2t1d0p0:boot - /boot pcfs - no -
```

要使用 Solaris Volume Manager 对根 (/) 文件系统进行镜像，该文件系统必须占用整个 Solaris fdisk 分区。因此，如果系统中已经存在 x86 引导分区，请使用 fdisk 命令删除该分区，然后重新安装 Solaris 软件。重新安装时，将不再重新创建引导分区。

注 - Solaris Volume Manager 只能对 Solaris fdisk 分区中的片进行镜像。如果有多个 fdisk 分区，则需要使用其他方法来保护 Solaris fdisk 分区以外的数据。

3 使辅助子镜像可以使用主引导程序引导。

a. 指定主引导程序。

```
# fdisk -b /usr/lib/fs/ufs/mboot /dev/rdisk/c1t1d0p0
```

此时屏幕将显示以下信息：

Total disk size is 31035 cylinders						
Cylinder size is 1146 (512 byte) blocks						
Cylinders						
Partition	Status	Type	Start	End	Length	%
=====	=====	=====	=====	=====	=====	=====
1	Active	Solaris	1	31034	31034	100

SELECT ONE OF THE FOLLOWING:

- 1. Create a partition
- 2. Specify the active partition
- 3. Delete a partition
- 4. Change between Solaris and Solaris2 Partition IDs
- 5. Exit (update disk configuration and exit)
- 6. Cancel (exit without updating disk configuration)

Enter Selection:

b. 从菜单中选择数字 5，然后按回车键。

4 使辅助磁盘可引导。

```
# /sbin/installgrub /boot/grub/stage1 /boot/grub/stage2 /dev/rdisk/c1t1d0s0
```

有关 installgrub 的更多信息，请参阅 installgrub(1M) 手册页。

5 确定包含要进行镜像的现有根 (/) 文件系统的片。

本示例使用 c1t0d0s0 片。

6 在上一步中的片上创建一个新的 RAID-0 卷。

RAID-0 卷中只能包含一个片。请使用以下方法之一：

- 从 Solaris Management Console 中增强的存储工具中，打开“卷”节点，然后选择“操作”⇒“创建卷”。请按照屏幕上的说明操作。有关更多信息，请参见联机帮助。
- 使用以下格式的 `metainit` 命令：

```
# metainit -f volume-name number-of-stripes components-per-stripe component-name
```

<code>-f</code>	强制继续执行该命令。如果片中包含已挂载的文件系统，则必须使用此选项。
<code>volume-name</code>	指定要创建的卷的名称。有关对卷进行命名的信息，请参见第 42 页中的“卷名称”。
<code>number-of-stripes</code>	指定要创建的条带的数目。
<code>components-per-stripe</code>	指定每个条带应包含的组件的数目。
<code>component-names</code>	指定所使用的组件的名称。本示例使用根片 <code>c0t0d0s0</code> 。

7 在未使用的片（在本示例中为 `clt1d0s0`）上创建另一个 RAID-0 卷，将其用作第二个子镜像。

注 - 辅助子镜像不得小于原来的子镜像的大小。另外，用作第二个子镜像的片还必须带有片标记“root”，并且根片必须是片 0。

有关配置片标记字段的信息，请参见 `format(1M)` 手册页。

请使用以下任一方法：

- 从 Solaris Management Console 中增强的存储工具中，打开“卷”节点，然后选择“操作”->“创建卷”并按照屏幕上的说明操作。有关更多信息，请参见联机帮助。
- 使用以下格式的 `metainit` 命令：

```
# metainit volume-name number-of-stripes components-per-stripes component-names
```

注 - 有关这些选项的说明，请参见步骤 6。

8 使用以下方法之一创建单向镜像：

- 从 Solaris Management Console 中增强的存储工具中，打开“卷”节点，然后选择“操作”⇒“创建卷”。请按照屏幕上的说明操作。有关更多信息，请参见联机帮助。
- 使用以下格式的 `metainit` 命令：

```
# metainit volume-name -m submirror-name
```

volume-name 指定要创建的卷的名称。

-m 指定要创建镜像。

submirror-name 指定将成为镜像中的第一个子镜像的组件的名称。在本示例中，这是包含根片的 RAID-0 卷。

9 重新挂载刚进行镜像的文件系统，然后重新引导系统。

```
# metaroot volume-name
```

```
# reboot
```

有关更多信息，请参见 `metaroot(1M)` 手册页。

10 附加第二个子镜像。

```
# metattach volume-name submirror-name
```

volume-name 指定要向其中添加子镜像的 RAID-1 卷的名称。

submirror-name 指定将成为附加到该镜像的第二个子镜像的组件的名称。

有关更多信息，请参见 `metattach(1M)` 手册页。

11 在 `menu.lst` 文件中定义替代引导路径。

要使系统可从用来存放辅助子镜像的磁盘引导，请将系统配置为将该磁盘视为备用引导设备。在当前示例中，替代路径 `c1t1d0s0` 位于第二块磁盘上第一个 `fdisk` 分区的第一片。因此，可以编辑 `menu.lst` 中的以下项：

```
title alternate boot
```

```
root (hd1,0,a)
```

```
kernel /boot/multiboot
```

```
module /boot/x86.miniroot-safe
```

注 – 要正确编辑 `menu.lst` 中的各项，必须熟悉 GRUB 中的磁盘命名约定。有关详细信息，请参见《系统管理指南：基本管理》中的第 11 章“基于 GRUB 的引导（任务）”。

完成编辑 `menu.lst` 文件之后，系统会设置为向第二块磁盘进行故障转移。如果主磁盘出现故障，则磁盘编号会更改，以便系统从辅助磁盘引导。



注意 - 在某些情况下，BIOS 的自动磁盘重新编号功能可能会对从不可用的主磁盘进行恢复造成影响。当磁盘重新编号功能强制系统从辅助磁盘引导时，主磁盘的引导归档会过时。如果该主磁盘以后变为可用，则在引导系统时，磁盘重新编号功能会再次切换到从缺省的主磁盘进行系统引导。但是，在这个阶段，主磁盘的引导归档仍然是过时的。因此，系统可能根本无法引导。所以在这种情况下，请确保从 GRUB 菜单中选择正确的项，以便从有效的根归档中引导系统。系统完成引导过程之后，请执行常规的 `metadevice` 维护，这会对主磁盘和辅助磁盘均进行同步，并将有效的引导归档恢复到主磁盘。

▼ x86: 如何使用 DCA 从根 (/) 文件系统创建 RAID-1 卷

1 验证是否可将 BIOS 引导设备的顺序配置为允许系统从镜像中的第二块磁盘引导。

启动内核之前，系统由只读存储器 (read-only-memory, ROM) 基本输入/输出系统 (Basic Input/Output System, BIOS) 控制，该系统是基于 x86 的系统上的固件接口。BIOS 与基于 SPARC 的系统上的引导 PROM 类似。除了 BIOS 的其他启动功能以外，它还负责查找要从中引导系统的正确设备，以及从允许系统自行引导的设备中装入主引导记录。通常可以将 BIOS 配置为选择要探测其引导记录的设备的顺序。此外，现在实现的大多数 BIOS 都允许将设备配置为自动向辅助子镜像进行故障转移。如果系统没有此功能，则主子镜像失败时，需要在系统引导过程中访问 BIOS，以便将系统重新配置为从辅助根片引导。有关如何配置 BIOS 中的设置的说明，请查阅 BIOS 的用户指南。

可以使用系统上的 DCA 来验证是否可以从多块磁盘引导。某些设备驱动程序配置为只能看到系统上的一个磁盘。

2 验证 `fdisk` 分区是否配置为支持根镜像。

基于 x86 的系统的另一个功能是可使用 `fdisk` 分区。使用 Solaris OS 安装程序缺省的引导磁盘分区布局可以创建两个分区，一个是 Solaris `fdisk` 分区，另一个是名为 x86 引导分区的 `fdisk` 分区，其大小大约为 10MB。

对根 (/) 文件系统进行镜像时，x86 引导分区会产生问题。x86 引导分区位于 Solaris `fdisk` 分区外部。因此，x86 引导分区不能由 Solaris Volume Manager 进行镜像。此外，由于仅存在 x86 引导分区的一个副本，因此它还表示一个单点故障。

可以确定 Solaris OS 是否有一个单独的 x86 引导分区。x86 引导分区会挂载到 `/etc/vfstab` 文件中，其中包含类似如下的项：

```
/dev/dsk/c2t1d0p0:boot - /boot pcfs - no -
```

如果不存在单独的 x86 引导分区，则该项不会出现在 `/etc/vfstab` 文件中。

要对根 (/) 文件系统进行镜像，需要自定义 `fdisk` 分区，以便删除 x86 引导分区并使用单个 Solaris `fdisk` 分区。如果要使用 Solaris Volume Manager 根镜像，请不要在系统安装过程中创建单独的 x86 引导分区。如果系统已经安装并且创建了一个单独的 x86 引导

分区，请使用 `fdisk` 命令删除该 `fdisk` 分区并重新安装系统。在安装过程中，请避免通过在安装过程中自定义磁盘分区来创建单独的 x86 引导分区。

注 – Solaris Volume Manager 只能对 Solaris `fdisk` 分区中的片进行镜像。如果有多个 `fdisk` 分区，则需要使用其他方法来保护 Solaris `fdisk` 分区以外的数据。

3 使辅助子镜像可以使用主引导程序引导。

a. 使用 `fdisk` 命令来指定主引导程序。

```
# fdisk -b /usr/lib/fs/ufs/mboot /dev/rdisk/c1t1d0p0
```

此时屏幕将显示以下信息：

Total disk size is 31035 cylinders

Cylinder size is 1146 (512 byte) blocks

Cylinders						
Partition	Status	Type	Start	End	Length	%
=====	=====	=====	=====	=====	=====	=====
1	Active	Solaris	1	31034	31034	100

SELECT ONE OF THE FOLLOWING:

- 1. Create a partition
- 2. Specify the active partition
- 3. Delete a partition
- 4. Change between Solaris and Solaris2 Partition IDs
- 5. Exit (update disk configuration and exit)
- 6. Cancel (exit without updating disk configuration)

Enter Selection:

b. 从菜单中选择数字 5，然后按回车键。

4 在辅助子镜像中安装引导块，以便可从该镜像引导系统。

辅助子镜像所在磁盘上的片 8 是从该 fdisk 分区引导 Solaris OS 所必需的。该片用来存放分区引导记录 (pboot)、磁盘的 Solaris VTOC 和引导块。由于这些信息特定于磁盘，因此不能使用 Solaris Volume Manager 进行镜像。但是，必须确保这两块磁盘都是可引导的，以便可以在主磁盘出现故障时从辅助磁盘引导。使用 `installboot` 命令可将第二块磁盘设置为 Solaris 可引导磁盘。有关更多信息，请参见 `installboot(1M)` 手册页。

必须将磁盘的片 2 指定为设备，并且片 2 必须包含整块磁盘。

```
# installboot /usr/platform/i86pc/lib/fs/ufs/pboot \
```

```
/usr/platform/i86pc/lib/fs/ufs/bootblk /dev/rdisk/clt1d0s2
```

5 确定包含要进行镜像的现有根 (/) 文件系统的片。本示例使用片 `clt0d0s0`。

6 使用以下方法之一，在上一步中的片上创建一个新的 RAID-0 卷。RAID-0 卷中只能包含一个片。

- 从 Solaris Management Console 中增强的存储工具中，打开“卷”节点，然后选择“操作”⇒“创建卷”。请按照屏幕上的说明操作。有关更多信息，请参见联机帮助。
- 使用以下格式的 `metainit` 命令：

```
# metainit -f volume-name number-of-stripes components-per-stripe component-name
```

`-f` 强制继续执行该命令。如果片中包含已挂载的文件系统，则必须使用此选项。

`volume-name` 指定要创建的卷的名称。有关对卷进行命名的信息，请参见第 42 页中的“卷名称”。

`number-of-stripes` 指定要创建的条带的数目。

`components-per-stripe` 指定每个条带应包含的组件的数目。

`component-names` 指定所使用的组件的名称。本示例使用根片 `c0t0d0s0`。

7 在未使用的片（在本示例中为 `clt1d0s0`）上创建另一个 RAID-0 卷，将其用作第二个子镜像。辅助子镜像不得小于原来的子镜像的大小。请使用以下方法之一：

注-用作第二个子镜像的片必须带有片标记“root”，并且根片必须是片 0。有关配置片标记字段的信息，请参见 `format(1M)` 手册页。

- 从 Solaris Management Console 中增强的存储工具中，打开“卷”节点，然后选择“操作”->“创建卷”并按照屏幕上的说明操作。有关更多信息，请参见联机帮助。

- 使用以下格式的 `metainit` 命令：

```
# metainit volume-name number-of-stripes components-per-stripes component-names
```

注 – 有关这些选项的说明，请参见步骤 6。

8 使用以下方法之一创建单向镜像：

- 从 Solaris Management Console 中增强的存储工具中，打开“卷”节点，然后选择“操作”⇒“创建卷”。请按照屏幕上的说明操作。有关更多信息，请参见联机帮助。
- 使用以下格式的 `metainit` 命令：

```
# metainit volume-name -m submirror-name
```

volume-name 指定要创建的卷的名称。

-m 指定要创建镜像。

submirror-name 指定将成为镜像中的第一个子镜像的组件的名称。在本示例中，这是包含根片的 RAID-0 卷。

9 重新挂载刚进行镜像的文件系统。运行 `metaroot volume-name` 命令，将 *volume-name* 替换为已创建的镜像的名称。然后重新引导系统。

```
# metaroot volume-name
```

```
# reboot
```

有关更多信息，请参见 `metaroot(1M)` 手册页。

10 使用以下格式的 `metattach` 命令附加第二个子镜像。

```
# metattach volume-name submirror-name
```

volume-name 指定要向其中添加子镜像的 RAID-1 卷的名称。

submirror-name 指定将成为附加到该镜像的第二个子镜像的组件的名称。

有关更多信息，请参见 `metattach(1M)` 手册页。

11 记录替代引导路径。

需要对系统进行配置，以便在主子镜像失败时，系统从辅助子镜像引导。要使系统可从用来存放辅助子镜像的磁盘引导，请将系统配置为将该磁盘视为备用引导设备。

- a. 确定备用引导设备的路径。针对附加作为根 (/) 文件系统镜像的第二个子镜像的片，使用 `ls -l` 命令。

```
# ls -l /dev/dsk/clt1d0s0

lrwxrwxrwx 1 root root 55 Mar 5 12:54 /dev/rdisk/clt1d0s0 -> ../
./devices/eisa/eha@1000,0/cmdk@1,0:a
```

- b. 记录 `/devices` 目录后面的字符串：`/eisa/eha@1000,0/cmdk@1,0:a`。这是设备树的路径。

注-由于系统可能不可用，因此应当将这些信息记录在系统以外的某个位置。这样，必须使用 DCA 引导系统时，可以更方便地输入设备树的路径信息。

- c. 使用 `eeeprom` 命令定义替代引导路径。例如：

```
# eeeprom altbootpath=/eisa/eha@1000,0/cmdk@1,0:a
```

如果主子镜像失败，系统会尝试从辅助子镜像引导。如果可将 BIOS 配置为自动故障转移到第二块磁盘，则引导过程是自动的。否则，需要进入 BIOS 并将其配置为从辅助磁盘引导。系统开始引导之后，即会尝试从 `bootpath` 设备引导。由于主引导磁盘是根镜像中的停用磁盘，因此系统随后会尝试从 `altbootpath` 设备引导。有关如何配置 BIOS 中的设置的说明，请查阅 BIOS 的用户指南。

如果系统不会自动引导，则可以尝试使用 DCA 来选择辅助子镜像。在某些系统上，可以选择在引导过程中进入 DCA。如果不提供此选项，则需要从 x86 引导软盘引导，并使用 DCA 来选择辅助子镜像。操作系统引导之后，请使用设置为替代引导路径的值（`bootpath` 值）来更新 `eeeprom bootpath` 值。然后，系统将自动引导。

有关使用 `eeeprom` 命令的更多信息，请参见 `eeeprom(1M)` 手册页。

示例 11-7 x86: 使用 DCA 从根 (/) 文件系统创建镜像

```
# metainit -f d1 1 1 c0t0d0s0
```

```
d1: Concat/Stripe is setup
```

```
# metainit d2 1 1 c0t1d0s0
```

```
d2: Concat/Stripe is setup
```

```
# metainit d0 -m d1
```

```
d0: Mirror is setup

# metaroot d0

# lockfs -fa

# reboot

...

# metattach d0 d2

d0: Submirror d2 is attached

# ls -l /dev/dsk/c0t1d0s0

lrwxrwxrwx  1 root    root          88 Feb  8 15:51 /dev/dsk/c1t3d0s0 ->
../../../../devices/pci@1f,0/pci@1,1/ide@3/dad@0,0:a,raw

# eeprom altbootpath=/pci@1f,0/pci@1,1/ide@3/dad@0,0:a,raw

# fdisk -b /usr/lib/fs/ufs/mboot /dev/dsk/c0t1d0p0

        Total disk size is 31035 cylinders

        Cylinder size is 1146 (512 byte) blocks
```

Cylinders						
Partition	Status	Type	Start	End	Length	%
=====	=====	=====	=====	=====	=====	=====
1	Active	Solaris	1	31034	31034	100

SELECT ONE OF THE FOLLOWING:

- 1. Create a partition

2. Specify the active partition
3. Delete a partition
4. Change between Solaris and Solaris2 Partition IDs
5. Exit (update disk configuration and exit)
6. Cancel (exit without updating disk configuration)

Enter Selection: 5

```
# installboot /usr/platform/i86pc/lib/fs/ufs/pboot \
/usr/platform/i86pc/lib/fs/ufs/bootblk /dev/rdisk/c0t1d0s2
```

了解对根 (/) 文件系统进行镜像时的引导时警告

对根 (/) 文件系统进行镜像之后，控制台中将会显示错误消息，这些错误消息将记录在 `/etc/syslog.conf` 文件中定义的系统日志中。这些错误消息并不表明存在问题。这些消息显示的是当前不能使用的每种设备类型，因为不能强制装入未使用的模块。错误消息如下所示：

```
Jul 13 10:17:42 ifr genunix: [ID 370176 kern.warning] WARNING: forceload of
misc/md_trans failed
```

```
Jul 13 10:17:42 ifr genunix: [ID 370176 kern.warning] WARNING: forceload of
misc/md_raid failed
```

```
Jul 13 10:17:42 ifr genunix: [ID 370176 kern.warning] WARNING: forceload of
misc/md_hotspares failed
```

可以放心地忽略这些错误消息。

处理子镜像

▼ 如何附加子镜像

注 – 指示 "can't attach labeled submirror to an unlabeled mirror" 的错误消息表示您未能成功尝试向镜像中附加 RAID-0 卷。带标号的卷（子镜像）的第一个组件从柱面 0 开始，而不带标号的卷的第一个组件则从柱面 1 开始。为了防止带标号的子镜像的标签损坏，Solaris Volume Manager 不允许将带标号的子镜像附加到不带标号的镜像。

开始之前 请阅读第 97 页中的“创建和维护 RAID-1 卷”。

1 确定要用作子镜像的组件（串联或条带）。

该组件不得小于镜像中现有子镜像的大小。如果尚未创建要成为子镜像的卷，请参见第 83 页中的“创建 RAID-0（条带化）卷”或第 85 页中的“创建 RAID-0（串联）卷”。

2 确保您具有超级用户权限，并且拥有所有数据的最新备份。

3 使用 `metastat` 命令验证要处理的镜像的状态是否为“Okay（正常）”。

```
# metastat mirror
```

4 使用以下方法之一来附加子镜像。

- 从 Solaris Management Console 中增强的存储工具中，打开“卷”节点。选择镜像。然后选择“操作”⇒“属性”并单击“子镜像”选项卡。请按照屏幕上的说明操作。有关更多信息，请参见联机帮助。
- 使用 `metattach mirror submirror` 命令。

```
# metattach mirror submirror
```

有关更多信息，请参见 `metattach(1M)` 手册页。

5 使用 `metastat` 命令查看镜像的状态。

```
# metastat mirror
```

示例 11-8 附加子镜像

```
# metastat d30
```

```
d30: mirror
```



```
Submirror 0: d60

State: Okay

...

# metattach d30 d70

d30: submirror d70 is attached

# metastat d30

d30: mirror

Submirror 0: d60

State: Okay

Submirror 1: d70

State: Resyncing

Resync in progress: 41 % done

Pass: 1

Read option: roundrobin (default)

Write option: parallel (default)

Size: 2006130 blocks

...
```

本示例说明如何将子镜像 **d70** 附加到单向镜像 **d30**。将子镜像附加到镜像时，即会创建一个双向镜像。镜像 **d30** 最初由子镜像 **d60** 组成。子镜像 **d70** 是 RAID-0 卷。可以使用 **metastat** 命令验证镜像的状态是否为“Okay（正常）”，然后再附加子镜像。运行 **metattach** 命令时，新的子镜像会与现有镜像重新同步。向镜像附加其他子镜像时，系统会显示一条消息。要验证镜像是否正在重新同步，请使用 **metastat** 命令。

▼ 如何拆离子镜像

开始之前 请阅读第 97 页中的“创建和维护 RAID-1 卷”。

- 1 确保您具有超级用户权限，并且拥有所有数据的最新备份。
- 2 使用 `metastat` 命令验证要处理的镜像的状态是否为“Okay (正常)”。
- 3 使用以下方法之一来拆离子镜像。
 - 从 Solaris Management Console 中增强的存储工具中，打开“卷”节点。选择镜像。然后选择“操作”⇒“属性”并单击“子镜像”选项卡。请按照屏幕上的说明操作。有关更多信息，请参见联机帮助。
 - 使用 `metadetach` 命令从镜像中拆离子镜像。

```
# metadetach mirror submirror
```

有关更多信息，请参见 `metadetach(1M)` 手册页。

示例 11-9 拆离子镜像

```
# metastat
```

```
d5: mirror
```

```
    Submirror 0: d50
```

```
...
```

```
# metadetach d5 d50
```

```
d5: submirror d50 is detached
```

在本示例中，镜像 `d5` 包含一个子镜像 `d50`。可以使用 `metadetach` 命令拆离子镜像。`d50` 中的基础片可以在其他位置重用。从镜像中拆离子镜像之后，系统会显示一条确认消息。

▼ 如何使子镜像脱机和联机

只有当已经使用 `metaoffline` 命令使子镜像脱机时，才能使用 `metaonline` 命令。运行 `metaonline` 命令之后，Solaris Volume Manager 会自动开始将子镜像与镜像重新同步。

注 - `metaoffline` 命令的功能与 `metadetach` 命令所提供的功能相似。但是，`metaoffline` 命令不会切断子镜像和镜像之间的逻辑关联。

开始之前 请阅读第 97 页中的“创建和维护 RAID-1 卷”。

- 1 确保您具有超级用户权限，并且拥有所有数据的最新备份。
- 2 使用以下方法之一使子镜像联机或脱机。
 - 从 Solaris Management Console 中增强的存储工具中，打开“卷”节点。选择镜像。然后选择“操作”⇒“属性”并单击“子镜像”选项卡。请按照屏幕上的说明操作。有关更多信息，请参见联机帮助。
 - 使用 `metaoffline` 命令使子镜像脱机。

```
# metaoffline mirror submirror
```

有关更多信息，请参见 `metaoffline(1M)` 手册页。

- 使用 `metaonline` 命令使子镜像联机。

```
# metaonline mirror submirror
```

有关更多信息，请参见 `metaonline(1M)` 手册页。

示例 11-10 使子镜像脱机

```
# metaoffline d10 d11
```

```
d10: submirror d11 is offlined
```

在本示例中，子镜像 `d11` 将从镜像 `d10` 中脱机。系统将继续从其他子镜像中进行读取。首次进行写入之后，镜像将不再同步。使脱机子镜像恢复联机后，即可更正这种不一致性。

示例 11-11 使子镜像联机

```
# metaonline d10 d11d10: submirror d11 is online
```

在本示例中，使镜像 `d10` 中的子镜像 `d11` 恢复联机。

▼ 如何启用子镜像中的片

开始之前 请阅读第 242 页中的“在 RAID-1 和 RAID-5 卷中替换和启用组件的概述”和第 97 页中的“创建和维护 RAID-1 卷”。

- 1 确保您具有超级用户权限，并且拥有所有数据的最新备份。
- 2 使用以下方法之一来启用子镜像中的片。
 - 从 Solaris Management Console 中增强的存储工具中，打开“卷”节点。选择镜像。然后选择“操作”⇒“属性”并单击“子镜像”选项卡。请按照屏幕上的说明操作。有关更多信息，请参见联机帮助。
 - 使用 `metareplace` 命令启用子镜像中出现故障的片。

```
# metareplace -e mirror failed-slice
```

`metareplace` 命令会自动开始重新同步，以便将已修复或替换的片与镜像的其余部分同步。

有关更多信息，请参见 `metareplace(1M)` 手册页。

示例 11-12 启用子镜像中的片

```
# metareplace -e d11 c1t4d0s7
```

```
d11: device c1t4d0s7 is enabled
```

在本示例中，镜像 `d11` 有一个包含片 `c1t4d0s7` 的子镜像，该片中有一个软错误。使用带有 `-e` 选项的 `metareplace` 命令可启用出现故障的片。

如果物理磁盘有缺陷，则可以使用系统中的另一块可用磁盘（及其片）来替换有缺陷的磁盘，如第 138 页中的“如何替换子镜像中的片”中所述。或者，也可以修理或替换磁盘，将其格式化，然后运行带有 `-e` 选项的 `metareplace` 命令，如本示例中所示。

维护 RAID-1 卷

▼ 如何查看镜像和子镜像的状态

开始之前 有关与 RAID-1 卷和子镜像相关联的状态信息的概述，请参见第 99 页中的“了解子镜像状态以确定维护操作”。

- 1 使用以下方法之一来检查镜像或子镜像的状态。

- 从 Solaris Management Console 中增强的存储工具中，打开“卷”节点。选择镜像。然后选择“操作”⇒“属性”。请按照屏幕上的说明操作。有关更多信息，请参见联机帮助。
- 针对镜像运行 `metastat` 命令以查看每个子镜像的状态。

`metastat mirror`

要更改镜像的传送号、读取选项或写入选项，请参见第 136 页中的“如何更改 RAID-1 卷选项”。

有关检查设备状态的更多信息，请参见 `metastat(1M)`。

示例 11-13 检查 RAID-1 卷的状态

以下是 `metastat` 命令的样例输出。使用不带镜像名称的 `metastat` 命令可以显示所有镜像的全部状态。

```
# metastatd70: Mirror

Submirror 0: d71

State: Okay

Pass: 1

Read option: roundrobin (default)

Write option: parallel (default)

Size: 12593637 blocks

d71: Submirror of d70

State: Okay

Size: 12593637 blocks

Stripe 0:

Device          Start Block  Dbase State      Reloc  Hot Spare

c1t3d0s3        0           No    Okay          Yes

Stripe 1:
```

```

Device          Start Block  Dbase State      Reloc  Hot Spare
c1t3d0s4          0      No   Okay        Yes

Stripe 2:

Device          Start Block  Dbase State      Reloc  Hot Spare
c1t3d0s5          0      No   Okay        Yes

d0: Mirror

Submirror 0: d1

State: Okay

Submirror 1: d2

State: Okay

Pass: 1

Read option: roundrobin (default)

Write option: parallel (default)

Size: 5600 blocks

d1: Submirror of d0

State: Okay

Size: 5600 blocks

Stripe 0:

Device          Start Block  Dbase State      Hot Spare
c0t2d0s7          0      No   Okay

...
```

使用带有镜像名称参数的 `metastat` 命令可以显示特定镜像的输出。

metastat d70

d70: Mirror

Submirror 0: d71

State: Okay

Pass: 1

Read option: roundrobin (default)

Write option: parallel (default)

Size: 12593637 blocks

d71: Submirror of d70

State: Okay

Size: 12593637 blocks

Stripe 0:

Device	Start Block	Dbase	State	Reloc	Hot Spare
c1t3d0s3	0	No	Okay	Yes	

Stripe 1:

Device	Start Block	Dbase	State	Reloc	Hot Spare
c1t3d0s4	0	No	Okay	Yes	

Stripe 2:

Device	Start Block	Dbase	State	Reloc	Hot Spare
c1t3d0s5	0	No	Okay	Yes	

对于镜像中的每个子镜像，`metastat` 命令会显示其状态、"invoke" 行（如果有错误）、所指定的热备用池（如果有）、大小（以块为单位）以及有关子镜像中每个片的信息。

▼ 如何更改 RAID-1 卷选项

开始之前 请查看第 98 页中的“关于 RAID-1 卷选项”。

- 1 确保您具有超级用户权限，并且拥有所有数据的最新备份。
- 2 使用以下方法之一来更改 RAID-1 选项。
 - 从 Solaris Management Console 中增强的存储工具中，打开“卷”节点。选择镜像。然后选择“操作”⇒“属性”。请按照屏幕上的说明操作。有关更多信息，请参见联机帮助。
 - 使用 `metaparam` 命令显示和更改镜像的选项。

```
# metaparam [mirror options] mirror
```

有关镜像选项的说明，请参见第 98 页中的“关于 RAID-1 卷选项”。另请参见 `metaparam(1M)` 手册页。

示例 11-14 更改 RAID-1 卷的读取策略

```
# metaparam -r geometric d30
```

```
# metaparam d30
```

```
d30: mirror current parameters are:
```

```
Pass: 1
```

```
Read option: geometric (-g)
```

```
Write option: parallel (default)
```

在本示例中，`-r` 选项用于将镜像的读取策略更改为 `geometric`。

示例 11-15 更改 RAID-1 卷的写入策略

```
# metaparam -w serial d40
```

```
# metaparam d40
```

```
d40: mirror current parameters are:
```

```
Pass: 1
```

```
Read option: roundrobin (default)
```



```
Write option: serial (-S)
```

在本示例中，`-w` 选项用于将镜像的写入策略更改为 `serial`。

示例 11-16 更改 RAID-1 卷的传送号

```
# metaparam -p 5 d50
```

```
# metaparam d50
```

```
d50: mirror current parameters are:
```

```
Pass: 5
```

```
Read option: roundrobin (default)
```

```
Write option: parallel (default)
```

在本示例中，`-p` 选项用于将镜像的传送号更改为 5。

▼ 如何扩展 RAID-1 卷

开始之前 请阅读第 97 页中的“创建和维护 RAID-1 卷”。

- 1 确保您具有超级用户权限，并且拥有所有数据的最新备份。
- 2 使用以下方法之一来扩展镜像。
 - 从 Solaris Management Console 中增强的存储工具中，打开“卷”节点。选择镜像。然后选择“操作”->“属性”并单击“子镜像”选项卡。请按照屏幕上的说明操作。有关更多信息，请参见联机帮助。
 - 使用 `metattach` 命令向每个子镜像附加其他片。

```
# metattach submirror slice
```

必须对镜像中的每个子镜像都进行扩展。有关更多信息，请参见 `metattach(1M)` 手册页。

- 3 使用 `metattach` 命令可以基于子镜像的大小重新计算镜像的大小。

```
# metattach mirror
```

示例 11-17 扩展包含已挂载文件系统的双向镜像

```
# metastat

d8: Mirror

    Submirror 0: d9

        State: Okay

    Submirror 1: d10

        State: Okay

...

# metattach d9 c0t2d0s5

d9: component is attached

# metattach d10 c0t3d0s5

d10: component is attached

# metattach d8
```

本示例说明如何通过将两个磁盘驱动器串联到镜像的两个子镜像来扩展已挂载的镜像文件系统。该镜像的名称为 **d8**，其中包含两个名为 **d9** 和 **d10** 的子镜像。

另请参见 对于 UFS，请针对镜像卷运行 `growfs(1M)` 命令。请参见第 241 页中的“[如何扩展文件系统](#)”。

使用原始卷的应用程序（如数据库）必须通过各自的方法来扩展已添加的存储器。

响应 RAID-1 卷组件的故障

▼ 如何替换子镜像中的片

开始之前 请阅读第 242 页中的“[在 RAID-1 和 RAID-5 卷中替换和启用组件的概述](#)”和第 97 页中的“[创建和维护 RAID-1 卷](#)”。

- 1 确保您具有超级用户权限，并且拥有所有数据的最新备份。

2 使用 `metastat` 命令查看 RAID-1 卷和关联子镜像的状态。

```
# metastat mirror-name
```

3 使用以下方法之一来替换子镜像中的片。

- 从 Solaris Management Console 中增强的存储工具中，打开“卷”节点。选择镜像。然后选择“操作”⇒“属性”并单击“子镜像”选项卡。请按照屏幕上的说明操作。有关更多信息，请参见联机帮助。
- 使用以下格式的 `metareplace` 命令替换子镜像中的片：

```
# metareplace mirror-name component-name
```

- *mirror-name* 是要创建的卷的名称。
- *component-name* 指定要替换的组件的名称。

mirror-name 指定要创建的卷的名称

component-name 指定要替换的组件的名称

有关更多信息，请参见以下示例和 `metainit(1M)` 手册页。

示例 11-18 替换镜像中出现故障的片

以下示例说明在未将系统配置为使用热备用池来自动替换出现故障的磁盘时，如何替换出现故障的片。有关使用热备用池的更多信息，请参见第 16 章。

```
# metastat d6
```

```
d6: Mirror
```

```
Submirror 0: d16
```

```
State: Okay
```

```
Submirror 1: d26
```

```
State: Needs maintenance
```

```
...
```

```
d26: Submirror of d6
```

```
State: Needs maintenance
```

```
Invoke: metareplace d6 c0t2d0s2 <new device>
```

```
...
```

```
# metareplace d6 c0t2d0s2 c0t2d2s2
```

```
d6: device c0t2d0s2 is replaced with c0t2d2s2
```

metastat 命令用于确认镜像 d6 是否包含一个子镜像 d26，并且该镜像中的一个片处于“Needs maintenance（需要维护）”状态。metareplace 命令会将在 metastat 输出的“Invoke”行中指定的片替换为系统中另一个可用的片。系统会确认该片是否已替换，并开始重新同步子镜像。

▼ 如何替换子镜像

开始之前 请阅读第 242 页中的“在 RAID-1 和 RAID-5 卷中替换和启用组件的概述”和第 97 页中的“创建和维护 RAID-1 卷”。

1 确保您具有超级用户权限，并且拥有所有数据的最新备份。

2 使用 metastat 命令查看 RAID-1 卷和关联子镜像的状态。

```
# metastat mirror-name
```

3 使用以下方法之一来替换子镜像。

- 从 Solaris Management Console 中增强的存储工具中，打开“卷”节点。选择镜像。然后选择“操作”⇒“属性”并单击“子镜像”选项卡。请按照屏幕上的说明操作。有关更多信息，请参见联机帮助。
- 使用 metadetach、metaclear、metatinit 和 metattach 命令替换整个子镜像。
 - a. 使用 metadetach 命令从镜像中拆离出故障的子镜像。

```
# metadetach -f mirror-name submirror
```

```
-f          强制进行拆离
```

```
mirror-name 指定镜像的名称
```

```
submirror   指定要拆离的子镜像
```

b. 使用 metaclear 命令删除子镜像。

```
# metaclear -f submirror
```

```
-f          强制删除子镜像
```

```
submirror   指定要删除的子镜像
```

c. 使用 metainit 命令创建新的子镜像。

```
# metainit volume-name number-of-stripes components-per-stripe component-name
```

- | | |
|------------------------------|---|
| <i>volume-name</i> | 指定要创建的卷的名称。有关对卷进行命名的信息，请参见第 42 页中的“卷名称”。 |
| <i>number-of-stripes</i> | 指定要创建的条带的数目。 |
| <i>components-per-stripe</i> | 指定每个条带应包含的组件的数目。 |
| <i>component-names</i> | 指定所使用的组件的名称。本示例使用根片 <code>c0t0d0s0</code> 。 |
- d. 使用 `metattach` 命令附加新的子镜像。

```
# metattach mirror submirror
```

示例 11-19 替换镜像中的子镜像

以下示例说明如何替换活动镜像中的子镜像。

```
# metastat d20

d20: Mirror

    Submirror 0: d21

        State: Okay

    Submirror 1: d22

        State: Needs maintenance

...

# metadetach -f d20 d22

d20: submirror d22 is detached

# metaclear -f d22

d22: Concat/Stripe is cleared

# metainit d22 2 1 c1t0d0s2 1 c1t0d1s2

d22: Concat/Stripe is setup

# metattach d20 d22

d20: components are attached
```

在本示例中，`metastat` 命令用于确认双向镜像 `d20` 是否包含处于“Needs maintenance（需要维护）”状态的子镜像 `d22`。在这种情况下，需要清除整个子镜像并重新创建。`metadetach` 命令通过使用 `-f` 选项来从镜像中拆离出故障的子镜像，该选项可强制进行拆离。`metaclear` 命令会清除子镜像。`metainit` 命令会使用新片重新创建 `d22` 子镜像。最后，`metattach` 命令会附加重新生成的子镜像。系统将开始自动重新同步镜像。

新卷 `d22` 的具体配置取决于要替换的组件。此处所示的串联足以替换条带。但是，使用串联来替换条带并不理想，因为它会影响性能。

如果是单向镜像，则会暂时失去数据冗余功能。

删除 RAID-1 卷（取消镜像）

▼ 如何取消镜像文件系统

对于可以在系统运行时取消挂载的文件系统，可以使用此过程对其取消镜像。要取消镜像根 (`/`) 文件系统、`/var` 文件系统、`/usr` 文件系统、`swap` 文件系统或者在系统运行时无法取消挂载的其他任何文件系统，请参见第 145 页中的“[如何取消镜像无法挂载的文件系统](#)”。

开始之前 请阅读第 97 页中的“[创建和维护 RAID-1 卷](#)”。

- 1 确保您具有超级用户权限，并且拥有所有数据的最新备份。

- 2 验证是否至少有一个子镜像处于 **Okay（正常）** 状态。

```
# metastat mirror
```

- 3 取消挂载文件系统。

```
# umount /file-system
```

- 4 拆离将继续用于文件系统的子镜像。

```
# metadetach mirror submirror
```

有关更多信息，请参见 `metadetach(1M)` 手册页。

- 5 清除镜像和剩余的子组件。

```
# metaclear -r mirror
```

有关更多信息，请参见 `metaclear(1M)` 手册页。

- 6 如有必要，请编辑 `/etc/vfstab` 文件，使其使用在 [步骤 4](#) 中拆离的组件。

7 重新挂载文件系统。

```
# mount /file-system
```

示例 11-20 取消镜像 /opt 文件系统

```
# metastat d4
```

```
d4: Mirror

    Submirror 0: d2

        State: Okay

    Submirror 1: d3

        State: Okay

    Pass: 1

    Read option: roundrobin (default)

    Write option: parallel (default)

    Size: 2100735 blocks (1.0 GB)


d2: Submirror of d4

    State: Okay

    Size: 2100735 blocks (1.0 GB)

    Stripe 0:

        Device      Start Block  Dbase      State Reloc Hot Spare

        c0t0d0s0      0           No          Okay    Yes


d3: Submirror of d4

    State: Okay
```

```
Size: 2100735 blocks (1.0 GB)

Stripe 0:

      Device      Start Block  Dbase      State Reloc Hot Spare

      c1t0d0s0      0      No      Okay    Yes
```

...

```
# umount /opt
```

```
# metadetach d4 d2
```

```
d4: submirror d2 is detached
```

```
# metaclear -r d4
```

```
d4: Mirror is cleared
```

```
d3: Concat/Stripe is cleared
```

（编辑 `/etc/vfstab` 文件使与 `/opt` 对应的项从 `d4` 更改为该基础片或卷）

```
# mount /opt
```

在本示例中，`/opt` 文件系统由双向镜像 `d4` 组成。该镜像的子镜像为 `d2` 和 `d3`。子镜像由片 `/dev/dsk/c0t0d0s0` 和 `/dev/dsk/c1t0d0s0` 组成。`metastat` 命令会确认是否至少有一个子镜像处于“Okay（正常）”状态。（必须首先修复其子镜像都不处于“Okay（正常）”状态的镜像。）文件系统将会取消挂载。然后，将拆离子镜像 `d2`。`metaclear -r` 命令会删除该镜像和另一个子镜像 `d3`。

接下来，会更改 `/etc/vfstab` 文件中与 `/opt` 相对应的项，使其引用基础片。

在本示例中，`/etc/vfstab` file 包含 `/opt` 文件系统的以下项：

```
/dev/md/dsk/d4 /dev/md/rdisk/d4 /opt ufs 2 yes -
```

请更改该项，使其显示为：

```
/dev/md/dsk/d2 /dev/md/rdisk/d2 /opt ufs 2 yes -
```

使用子镜像名称，可以使文件系统继续挂载在卷上。最后，将重新挂载 `/opt` 文件系统。

在 `/etc/vfstab` 文件中使用 `d2` 而不是 `d4`，可以取消镜像。由于 `d2` 由一个片组成，因此，如果不希望设备支持卷，则可以在该片（其名称为 `/dev/dsk/c0t0d0s0`）上挂载文件系统。

▼ 如何取消镜像无法挂载的文件系统

使用此任务，可以取消镜像在正常系统运行过程中无法取消挂载的文件系统，包括根 (`/`) 文件系统、`/usr` 文件系统、`/opt` 文件系统和 `swap` 文件系统。

- 1 确保您具有超级用户权限，并且拥有所有数据的最新备份。

- 2 验证是否至少有一个子镜像处于 **Okay**（正常）状态。

```
# metastat mirror
```

- 3 拆离将继续用于文件系统的子镜像。

```
# metadetach mirror submirror
```

有关更多信息，请参见 `metadetach(1M)` 手册页。

- 4 根据要取消镜像的文件系统，使用以下命令之一：

- 对于 `/usr`、`/opt` 或 `swap` 文件系统，更改 `/etc/vfstab` 文件中与各文件系统相对应的项，使其使用非 Solaris Volume Manager 设备（片）。
- 仅限根 (`/`) 文件系统：运行 `metaroot` 命令。

```
# metaroot rootslice
```

有关更多信息，请参见 `metaroot(1M)` 手册页。

- 5 重新引导系统。

```
# reboot
```

- 6 清除剩余的镜像和子镜像。

```
# metaclear -r mirror
```

有关更多信息，请参见 `metaclear(1M)` 手册页。

示例 11-21 取消镜像根 (`/`) 文件系统

```
# metastat d0
```

```
d0: Mirror
```

Submirror 0: d10

State: Okay

Submirror 1: d20

State: Okay

Pass: 1

Read option: roundrobin (default)

Write option: parallel (default)

Size: 2100735 blocks (1.0 GB)

d10: Submirror of d0

State: Okay

Size: 2100735 blocks (1.0 GB)

Stripe 0:

Device	Start Block	Dbase	State	Reloc	Hot Spare
c0t3d0s0	0	No	Okay	Yes	

d20: Submirror of d0

State: Okay

Size: 2100735 blocks (1.0 GB)

Stripe 0:

Device	Start Block	Dbase	State	Reloc	Hot Spare
c1t3d0s0	0	No	Okay	Yes	

```
# metadetach d0 d20

d0: submirror d20 is detached

# metaroot /dev/dsk/c0t3d0s0

# reboot

...

# metaclear -r d0

d0: Mirror is cleared

d10: Concat/Stripe is cleared

# metaclear d20

d20: Concat/Stripe is cleared
```

在本示例中，根 (/) 文件系统是双向镜像 **d0**。该镜像的子镜像为 **d10** 和 **d20**。子镜像由片 **/dev/dsk/c0t3d0s0** 和 **/dev/dsk/c1t3d0s0** 组成。**metastat** 命令会确认是否至少有一个子镜像处于“Okay（正常）”状态。（必须首先修复其子镜像都不处于“Okay（正常）”状态的镜像。）将拆离子镜像 **d20**，使 **d0** 成为单向镜像。

rootslice 是包含根 (/) 文件系统的片。**metaroot** 命令会使用要从中引导系统的 **rootslice** 来运行。此命令会编辑 **/etc/system** 和 **/etc/vfstab** 文件。此命令会删除用来指定对根 (/) 文件系统进行镜像的信息。

系统重新引导之后，**metaclear -r** 命令会删除该镜像和另一个子镜像 **d10**。最后一个 **metaclear** 命令会清除子镜像 **d20**。

示例 11-22 取消镜像 swap 文件系统

```
# metastat d1

d1: Mirror

    Submirror 0: d11

        State: Okay

    Submirror 1: d21
```

```

State: Okay

...

# metadetach d1 d21

d1: submirror d21 is detached

（编辑 /etc/vfstab 文件以将与 swap 对应的项从 metadvice 更改为片名称）

# reboot

...

# metaclear -r d1

d1: Mirror is cleared

d11: Concat/Stripe is cleared

# metaclear d21

d21: Concat/Stripe is cleared

```

在本示例中，swap 文件系统由双向镜像 d1 组成。该镜像的子镜像为 d11 和 d21。子镜像由片 /dev/dsk/c0t3d0s1 和 /dev/dsk/c1t3d0s1 组成。metastat 命令会确认是否至少有一个子镜像处于“Okay（正常）”状态。（必须首先修复其子镜像都不处于“Okay（正常）”状态的镜像。）将拆离子镜像 d21，使 d1 成为单向镜像。接下来，会编辑 /etc/vfstab 文件，以更改与 swap 相对应的项，使其引用子镜像 d21 中的片。

在本示例中，/etc/vfstab file 包含 swap 文件系统的以下项：

```

/dev/md/dsk/d4  /dev/md/rdisk/d4  /opt ufs  2  yes  -

/dev/md/dsk/d1  - - swap  - no  -

```

请更改该项，使其显示为：

```

/dev/dsk/c0t3d0s1  - - swap  - no  -

```

系统重新引导之后，metaclear -r 命令会删除该镜像和另一个子镜像 d11。最后一个 metaclear 命令会清除子镜像 d21。

在 RAID-1 卷上备份数据

Solaris Volume Manager 并不是“备份产品”。Solaris Volume Manager 确实提供了一种备份镜像数据的方法，而不会导致出现以下任一情况：

- 取消挂载镜像
- 使整个镜像脱机
- 停止系统
- 拒绝用户访问数据

Solaris Volume Manager 通过首先使其中一个子镜像脱机来备份镜像数据。在备份过程中，镜像将暂时不可用。一旦备份完成，子镜像将立即恢复联机和重新同步。

注 - UFS 快照功能提供了一种不使文件系统脱机即可备份系统的备选方法。可以在不拆离子镜像的情况下执行备份，这样也避免了以后由于重新同步镜像而导致性能损失。使用 UFS 快照功能执行备份之前，请确保 UFS 文件系统中有足够的可用空间。有关更多信息，请参见《系统管理指南：设备和文件系统》中的第 26 章“使用 UFS 快照（任务）”。

▼ 如何对 RAID-1 卷执行联机备份

可以针对除根 (/) 文件系统以外的任何文件系统使用此过程。请注意，这种类型的备份会创建活动文件系统的“快照”。根据文件系统处于写入锁定状态时的使用方式，所备份的某些文件可能与磁盘上的实际文件不对应。

以下限制适用于此过程：

- 如果针对双向镜像使用此过程，请注意，在使一个子镜像脱机以进行备份时，会失去数据冗余功能。多向镜像则不存在此问题。
- 备份完成之后，对重新附加的子镜像重新同步时，会产生一些系统开销。

此过程中的概括性步骤如下所示：

- 对文件系统进行写入锁定（仅限 UFS）。请勿锁定根 (/) 文件系统
- 将高速缓存中的所有数据刷新到磁盘。
- 使用 `metadetach` 命令从镜像中拆离某个子镜像
- 解除锁定文件系统
- 使用 `fsck` 命令检查已拆离子镜像中的文件系统
- 将数据备份到已拆离子镜像中
- 使用 `metattach` 命令将已拆离子镜像重新放回镜像中

注 - 如果要定期使用上述过程，请将其置入脚本中以方便使用。

提示 – 对于此过程，比较安全的做法是向镜像中附加第三个或第四个子镜像，允许该子镜像重新同步，然后使用该子镜像进行备份。此方法可确保一直保持数据冗余功能。

1 验证镜像是否处于“Okay（正常）”状态。

应当首先修复处于“Maintenance（维护）”状态的镜像。

```
# metastat mirror
```

2 将高速缓存中的数据和 UFS 日志记录数据刷新到磁盘上，并对文件系统进行写入锁定。

```
# /usr/sbin/lockfs -w mount-point
```

仅有 UFS 卷才需要进行写入锁定。如果将该卷设置为数据库管理软件或其他某个应用程序的原始设备，则不必运行 `lockfs` 命令。但是，您可能希望运行由供应商提供的适当实用程序来刷新所有缓冲区并锁定访问。



注意 – 请勿对根 (/) 文件系统进行写入锁定。对根 (/) 文件系统进行写入锁定会导致系统挂起。如果要备份根 (/) 文件系统，请跳过此步骤。

3 从镜像中拆离某个子镜像。

```
# metadetach mirror submirror
```

mirror 镜像的卷名。

submirror 拆离的子镜像（卷）的卷名。

系统将继续从其他子镜像中进行读取。首次进行写入之后，镜像将不再同步。按照[步骤 7](#)中的说明重新附加已拆离子镜像后，即可更正这种不一致性。

4 解除锁定文件系统，从而允许继续进行写入。

```
# /usr/sbin/lockfs -u mount-point
```

您可能需要基于在[步骤 2](#)中使用的与供应商有关的实用程序来执行必要的解除锁定过程。

5 使用 `fsck` 命令检查已拆离子镜像中的文件系统。此步骤可确保进行完全备份。

```
# fsck /dev/md/rdisk/name
```

6 对脱机的子镜像执行备份。

使用 `ufsdump` 命令或常用的备份实用程序。有关使用 `ufsdump` 命令执行备份的信息，请参见第 331 页中的“[使用 `ufsdump` 命令对已挂载的文件系统执行备份](#)”。

注 – 要确保备份正确，请使用**原始**卷名，如 `/dev/md/rdisk/d4`。使用原始卷名可以访问大于 2 GB 的存储器。

7 附加该子镜像。

```
# metattach mirror submirror
```

Solaris Volume Manager 会自动开始将该子镜像与相应的镜像重新同步。

示例 11-23 对 RAID-1 卷执行联机备份

本示例使用镜像 `d1`。该镜像由子镜像 `d2`、`d3` 和 `d4` 组成。子镜像 `d3` 将在 `d2` 和 `d4` 子镜像保持联机时进行拆离和备份。该镜像中的文件系统是 `/home1`。

```
# metastat d1
```

```
d1: Mirror
```

```
    Submirror 0: d2
```

```
        State: Okay
```

```
    Submirror 1: d3
```

```
        State: Okay
```

```
    Submirror 1: d4
```

```
        State: Okay
```

```
...
```

```
# /usr/sbin/lockfs -w /home1
```

```
# metadetach d1 d3
```

```
# /usr/sbin/lockfs -u /home1
```

```
# /usr/sbin/fsck /dev/md/rdisk/d3
```

(Perform backup using /dev/md/rdisk/d3)

```
# metattach d1 d3
```


软分区（概述）

本章提供有关 Solaris Volume Manager 软分区的信息。有关相关任务的信息，请参见第 13 章。

本章包含以下信息：

- 第 153 页中的“软分区概述”
- 第 154 页中的“软分区的配置指南”

软分区概述

随着磁盘存储容量的增加，磁盘阵列为 Solaris 系统提供了更大的逻辑设备。为了创建更多便于管理的文件系统或分区大小，用户可能需要将磁盘或逻辑卷划分为八个以上的分区。Solaris Volume Manager 的软分区功能可以满足此需要。

Solaris Volume Manager 最多可支持每个磁盘集 8192 个逻辑卷。此数字包括本地（即未指定的）磁盘集。Solaris Volume Manager 可以在需要卷时动态配置卷。

可以使用软分区根据需要将磁盘片或逻辑卷分成许多分区。必须为每个分区（即**软分区**）提供名称，就像为其他存储卷（如条带或镜像）提供名称一样。软分区命名后，只要它不包括在其他卷中，则应用程序（包括文件系统）便可对其进行访问。软分区包括在卷中后，就不应对其直接访问。

可以将软分区直接置于磁盘片上方，或置于镜像卷、条带化卷或 RAID-5 卷的顶部。软分区不能位于其他卷的上方和下方。例如，如果在一个软分区上生成了镜像，则不允许在条带上生成该软分区。

对于文件系统和其他应用程序，软分区看上去是单个连续逻辑卷。但是，软分区实际上是由一系列可能位于基础介质上的任意位置的**扩展**组成。除了软分区以外，磁盘上的分区扩展表头（也称为**系统恢复数据区域**）还记录有关软分区的信息，从而便于在出现灾难性系统故障时进行恢复。

软分区的配置指南

- 不能将用于软分区的片用于其他目的。
- 如果对磁盘分区或在产生的片上生成文件系统，则以后扩展片时就会修改或破坏磁盘格式。软分区的行为方式是不同的。可以扩展软分区，直到达到基础设备上的空间量，而不会移动或破坏其他软分区上的数据。
- 尽管从技术上说可以将软分区的扩展手动置于磁盘上的任意位置，但应该允许系统自动放置扩展。有关手动放置的扩展的示例，请参见第 228 页中的“[查看 Solaris Volume Manager 配置](#)”中 `metastat` 命令的输出。
- 可以在任何片上生成软分区。但是，创建占用整个磁盘的单个片，然后在该片上创建软分区是在磁盘级别上使用软分区的最有效方式。
- 软分区的最大大小受限于用于生成该软分区的片或逻辑卷的大小。由于此限制，应该在磁盘片顶部生成卷，然后在该卷顶部生成软分区。此策略允许您以后将组件添加到卷，然后根据需要扩展软分区。
- 为达到最大程度的灵活性和高可用性，请在磁盘片上生成 RAID-1（镜像）卷或 RAID-5 卷，然后在该镜像卷或 RAID-5 卷上创建软分区。

方案一软分区

软分区提供用于将较大存储空间划分为更多便于管理的空间的工具。例如，在其他方案（第 101 页中的“[方案一RAID-1 卷（镜像）](#)”或第 167 页中的“[方案一RAID-5 卷](#)”）中，大的存储集合提供许多 GB 的冗余存储。但是，许多可能的方案最初不需要如此多的空间。软分区使您能够将存储空间划分为更多便于管理的分区。其中每个分区都可以具有完整的文件系统。例如，可以在 RAID-1 卷或 RAID-5 卷的顶部创建 1000 个软分区，以便每个用户都可以在单独的文件系统上具有起始目录。如果用户需要更多空间，可能只需扩展软分区即可。

软分区（任务）

本章提供有关执行与 Solaris Volume Manager 软分区相关联的任务的信息。有关与软分区有关的概念性信息，请参见第 12 章。

软分区（任务列表）

以下任务列表列出了管理 Solaris OS Volume Manager 软分区所需的过程。

任务	说明	参考
创建软分区	使用 Solaris Volume Manager GUI 或 <code>metainit</code> 命令创建软分区。	第 155 页中的“如何创建软分区”
检查软分区的状态	使用 Solaris Volume Manager GUI 或 <code>metastat</code> 命令检查软分区的状态。	第 157 页中的“如何检查软分区的状态”
扩展软分区	使用 Solaris Volume Manager GUI 或 <code>metattach</code> 命令扩展软分区。	第 158 页中的“如何扩展软分区”
删除软分区	使用 Solaris Volume Manager GUI 或 <code>metaclear</code> 命令删除软分区。	第 159 页中的“如何删除软分区”

创建软分区

▼ 如何创建软分区

开始之前 请查看第 154 页中的“软分区的配置指南”。

- 使用以下方法之一创建软分区：

- 从 Solaris Management Console 中增强的存储工具中，打开“卷”节点。选择“操作” ⇒ “创建卷”，然后按照向导中的说明操作。有关更多信息，请参见联机帮助。
- 要创建软分区，请使用 `metainit` 命令的以下格式：

```
# metainit [-s diskset] soft-partition -p [-e] component size
```

<code>-sdiskset</code>	指定要使用的磁盘集。如果未指定 <code>-s</code> ，则将使用本地（缺省）磁盘集。
<code>-p</code>	指定要对软分区进行配置。
<code>-e</code>	指定应当重新格式化整个磁盘。格式化磁盘会提供片 0，该片占据磁盘的大部分空间。格式化磁盘还会提供大小至少为 4 MB 的片 7，该片包含状态数据库副本。
<code>soft-partition</code>	指定软分区的名称。该名称采用 <code>dnnn</code> 格式，其中， <code>nnn</code> 是介于 0 和 8192 之间的数字。
<code>component</code>	指定要创建软分区的磁盘、片或逻辑卷。 该部件上现有的所有数据都将销毁 ，因为软分区头写在该部件的开头。
<code>size</code>	指定软分区的大小。软分区的大小用数字指定，单位可以是以下之一： <ul style="list-style-type: none"> ■ M 或 m（兆字节） ■ G 或 g（千兆字节） ■ T 或 t（兆兆字节） ■ B 或 b（块（扇区））

有关更多信息，请参见以下示例和 `metainit(1M)` 手册页。

示例 13-1 创建软分区

以下示例将在 `c1t3d0s2` 上创建一个名为 `d20` 的 4 GB 软分区。

```
# metainit d20 -p c1t3d0s2 4g
```

示例 13-2 获取软分区所在的整块磁盘

以下示例创建一个软分区并格式化磁盘 `c1t2d0`。此操作会销毁该磁盘上的所有数据并在片 0 上创建一个新的软分区。

```
# metainit d7 -p -e c1t2d0 1G
```

维护软分区

维护软分区与维护其他逻辑卷没有什么不同。

▼ 如何检查软分区的状态

开始之前 请阅读第 154 页中的 “软分区的配置指南”。

- 可使用以下方法之一检查软分区的状态：
 - 从 Solaris Management Console 中增强的存储工具中，打开“卷”节点。选择要监视的软分区，然后选择“操作”⇒“属性”。请按照屏幕上的说明操作。有关更多信息，请参见联机帮助。
 - 要查看现有的配置，请使用 `metastat` 命令的以下格式：

```
# metastat soft-partition
soft-partition 指定要检查的分区的名称。
```

示例 13-3 检查软分区的状态

以下示例将检查软分区 `d1` 的状态。此软分区包括两个扩展盘区，并且是在 RAID-1 卷 `d100` 上构建的。

```
# metastat d1

d1: soft partition

    component:  d100

    state: OKAY

    size:  42674285 blocks

            Extent          Start Block          Block Count

            0                10234                40674285

            1                89377263             20000000

d100: Mirror

    Submirror 0: d10

    State: OKAY
```

Read option: roundrobin (default)

Write option: parallel (default)

Size: 426742857 blocks

d10: Submirror of d100

State: OKAY

Hot spare pool: hsp002

Size: 426742857 blocks

Stripe 0: (interlace: 32 blocks)

Device	Start Block	Dbase State	Hot Spare
c3t3d0s0	0	No	Okay

▼ 如何扩展软分区

如果某个软分区上未曾构建其他逻辑卷，则可以为该软分区增加空间。系统将查找可用空间并用以扩展分区。不会移动现有数据。

注 - 如果已经使用某个软分区创建了另一个卷（例如，如果它是 RAID-0 卷的部件），则不能对该软分区进行扩展。在大多数情况下，这可以通过在包含该软分区的设备上串联其他卷来为该设备提供更多的空间。有关更多信息，请参见第 86 页中的“扩展存储容量”。

开始之前 请阅读第 154 页中的“软分区的配置指南”。

► 使用以下方法之一扩展软分区：

- 从 Solaris Management Console 中增强的存储工具中，打开“卷”节点。选择要扩展的软分区，然后选择“操作”⇒“属性”。请按照屏幕上的说明操作。有关更多信息，请参见联机帮助。
- 要为软分区增加空间，请使用 metattach 命令的以下格式：

```
# metattach [-s diskset] soft-partition size
```

diskset 指定软分区所在的磁盘集的名称。

soft-partition 指定现有软分区的名称。

size 指定要添加的存储空间量。

示例 13-4 扩展软分区

以下示例说明如何向软分区中附加空间。文件系统随后会在软分区处于联机状态且已挂载时，使用 **growfs** 命令进行扩展。

```
# mount /dev/md/dsk/d20 /home2

# metattach d20 10g

# growfs -M /home2 /dev/md/rdisk/d20
```

有关 growfs 命令的更多信息，请参见第 240 页中的“使用 growfs 命令扩展文件系统”。

▼ 如何删除软分区

开始之前 请阅读第 154 页中的“软分区的配置指南”。

■ 使用以下方法之一删除软分区：

- 从 Solaris Management Console 中增强的存储工具中，打开“卷”节点。选择要删除的软分区，然后选择“操作”⇒“属性”。请按照屏幕上的说明操作。有关更多信息，请参见联机帮助。
- 要删除软分区，请使用 metaclear 命令的以下格式之一：

```
# metaclear [-s diskset] component

# metaclear [-s diskset] -r soft-partition

# metaclear [-s diskset] -p component
```

diskset 指定软分区所在的磁盘集。

soft-partition 指定要删除的软分区。

-r 指定要以递归方式删除逻辑卷，而不删除其他部件所依赖的卷。

-p 指定要清除指定部件上除打开的软分区之外的所有软分区。

component 指定要清除所有软分区的部件。

示例 13-5 删除软分区

以下示例说明如何删除 `c1t4d2s0` 上的所有软分区。

```
# metaclear -p c1t4d2s0
```


RAID-5 卷（概述）

本章提供有关 Solaris Volume Manager 的 RAID-5 卷的概念性信息。有关执行相关任务的信息，请参见第 15 章。

本章包含以下主题：

- 第 161 页中的 “RAID-5 卷的概述”
- 第 164 页中的 “创建 RAID-5 卷的背景信息”
- 第 165 页中的 “有关检查 RAID-5 卷状态的概述”
- 第 167 页中的 “有关替换和启用 RAID-5 卷中的片的概述”
- 第 167 页中的 “方案—RAID-5 卷”

RAID-5 卷的概述

RAID 级别 5 类似于条带化，但其奇偶校验数据分布于所有组件（磁盘或逻辑卷）中。如果某个组件出现故障，则可以根据其他组件上分布的数据和奇偶校验信息来重新生成该组件上的数据。在 Solaris Volume Manager 中，**RAID-5 卷**是支持 RAID 级别 5 的卷。

RAID-5 卷使用相当于卷中的一个组件的存储容量来存储冗余信息（奇偶校验）。此奇偶校验信息包含有关在 RAID-5 卷的其余组件上所存储用户数据的信息。也就是说，如果您有三个组件，则相当于有一个组件的存储容量将用于存储奇偶校验信息。如果您有五个组件，则其中仍有相当于一个组件的存储容量将用于存储奇偶校验信息。奇偶校验信息分布于卷的所有组件中。与镜像类似，RAID-5 卷也可以增强数据可用性，而且所需的硬件成本最低，对写操作性能的影响也是可接受的。但是，您不能对根 (/)、/usr 和 swap 文件系统使用 RAID-5 卷，也不能对其他现有文件系统使用 RAID-5 卷。

替换现有组件时，Solaris Volume Manager 会自动对 RAID-5 卷进行重新同步。如果出现系统故障或崩溃，Solaris Volume Manager 也会在重新引导期间重新同步 RAID-5 卷。

示例—RAID-5 卷

图 14-1 说明了包含四个磁盘（组件）的 RAID-5 卷。

前三个数据段将写入到组件 A（交错 1）、组件 B（交错 2）和组件 C（交错 3）中。接下来写入的数据段为奇偶校验段。此奇偶校验段将写入组件 D（P 1-3）中。此段包含前三个数据段的异或结果。接下来的三个数据段将写入组件 A（交错 4）、组件 B（交错 5）和组件 D（交错 6）中。然后，另一个奇偶校验段写入组件 C（P 4-6）中。

数据段和奇偶校验段的这种写入模式将使数据和奇偶校验分布在 RAID-5 卷的所有磁盘中。可以分别读取每个磁盘。奇偶校验可以防止出现单个磁盘故障。如果本示例中的每一张磁盘容量都为 2 GB，则 RAID-5 卷的总容量将为 6 GB。相当于一个磁盘的空间将被分配给奇偶校验。

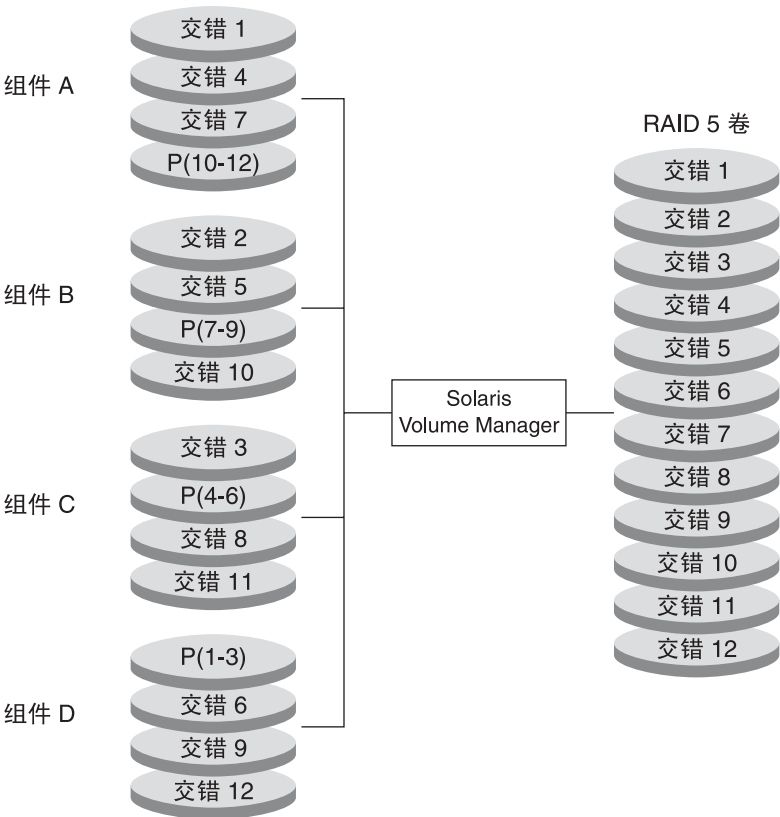


图 14-1 RAID-5 卷示例

示例—串联（扩展）RAID-5 卷

下图显示最初包含四张磁盘（组件）的 RAID-5 卷的示例。为扩展 RAID-5 卷，已将第五张磁盘与该磁盘动态串联。

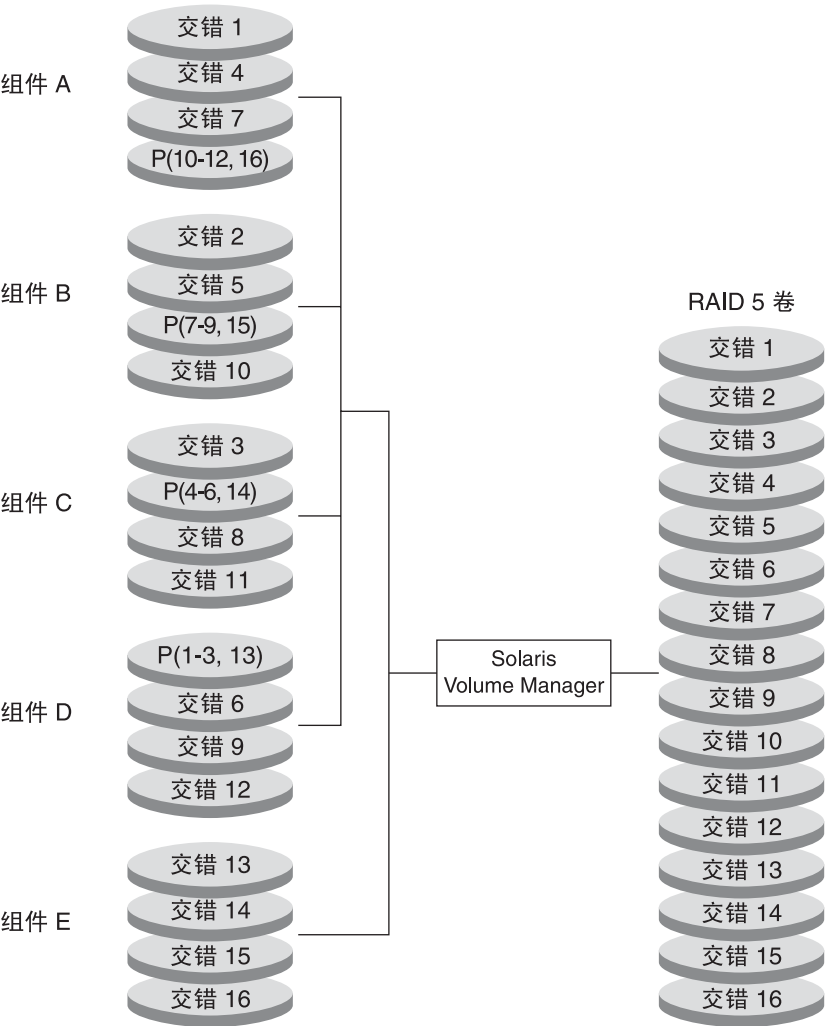


图 14-2 扩展的 RAID-5 卷示例

创建初始的 RAID-5 卷时会分配奇偶校验区域。尽管实际的奇偶校验块分布于用于分布 I/O 的所有原始组件中，但系统会为奇偶校验分配相当于一个组件的空间。将其他组件与 RAID-5 卷串联时，新增的空间将全部用于存储数据，不会分配任何新的奇偶校验块。但是，串联组件上的数据包括在奇偶校验计算中，因此可以保护数据不受单个设备故障的影响。

串联的 RAID-5 卷不适合长期使用。在重新配置更大 RAID-5 卷之前，可以使用串联的 RAID-5 卷。然后，将数据复制到更大的卷中。

注 - 将新的组件添加到 RAID-5 卷时，Solaris Volume Manager 会将该组件中的所有块都设为“零”。此过程确保奇偶校验能保护新数据。数据写入到新增的空间时，Solaris Volume Manager 会将这些数据包括在奇偶校验计算中。

创建 RAID-5 卷的背景信息

使用 RAID-5 卷时，请考虑第 164 页中的“RAID-5 卷的要求”和第 164 页中的“RAID-5 卷指南”。许多条带化原则也适用于 RAID-5 卷配置。请参见第 81 页中的“RAID-0 卷要求”。

RAID-5 卷的要求

- RAID-5 卷必须至少包含三个组件。不过，RAID-5 卷包含的组件越多，组件出现故障时读写操作所用的时间就越长。
- 不能对 RAID-5 卷执行条带化、串联或镜像操作。
- 请勿使用包含现有文件系统的组件来创建 RAID-5 卷。这样做将使数据在 RAID-5 初始化过程被清除。
- 创建 RAID-5 卷时，可以定义交错值。如果未指定交错值，则该值缺省为 16 KB。该值对于大多数应用程序都是合理的。
- RAID-5 卷（不含热备件）只能处理单个组件故障。
- 创建 RAID-5 卷时，请使用分布于不同控制器的组件。与磁盘相比，控制器及其关联电缆出现故障的频率往往会更频繁。
- 请使用大小相同的组件。使用大小不同的组件创建 RAID-5 卷会导致磁盘空间无法使用。

RAID-5 卷指南

- 由于奇偶校验计算的复杂性，写入容量超过 20% 的卷可能不应作为 RAID-5 卷。如果需要在包含大量写入内容的卷上具备数据冗余，可考虑镜像。
- 如果 RAID-5 卷中的不同组件驻留在不同的控制器上，而且对该卷的访问主要是大型连续访问，则将交错值设置为 32 KB 可能会提高性能。
- 可通过将其他组件与 RAID-5 卷串联起来扩展该卷。将新的组件与现有 RAID-5 卷串联可能会降低卷的总体性能，因为串联的数据是连续的。未跨越所有组件对数据进行条带化。卷的原始组件将跨越所有组件对数据和奇偶校验进行条带化。串联组件

中没有这种条带化。但是，由于在组件 I/O 期间使用奇偶校验，因此出现错误后仍可恢复数据。产生的 RAID-5 卷将继续处理单个组件故障。

此外，串联组件的不同之处还在于它们不会对任何区域进行奇偶校验条带化。因此，组件的全部内容都可用于数据。

串联组件时，大型或连续写入操作的任何性能增强都将丧失。

- 您可以创建 RAID-5 卷，而不必将数据块都“设置为零”。为此，请执行以下操作之一：
 - 使用带有 -k 选项的 `metainit` 命令。-k 选项将重新创建 RAID-5 卷，而不对其进行初始化，而且会将磁盘块设置为“Okay（正常）”状态。该选项可能很危险，因为卷内的磁盘块上存在的任何错误都将导致 Solaris Volume Manager 中出现不可预测的行为，包括伪造数据的可能性。
 - 请初始化设备并从磁带中恢复数据。有关更多信息，请参见 `metainit(1M)` 手册页。

有关检查 RAID-5 卷状态的概述

通过查看 RAID-5 卷状态和卷的片状态，可以检查这些卷的状态。排除 RAID-5 卷错误时，片状态可提供最具体的信息。RAID-5 卷状态仅提供常规状态信息，如“Okay（正常）”或“Maintenance（维护）”。

如果 RAID-5 卷状态报告“Maintenance（维护）”状态，请参阅片状态。片状态具体报告片是处于“Maintenance（维护）”状态，还是处于“Last Erred（最近出错）”状态。根据片处于“Maintenance（维护）”状态还是处于“Last Erred（最近出错）”状态，可以执行不同的恢复操作。如果只具有处于“Maintenance（维护）”状态的片，则可以对片进行修复，而不会丢失数据。如果同时即有处于“Maintenance（维护）”状态的片，又有处于“Last Erred（最近出错）”状态的片，则数据可能已被损坏。必须首先修复处于“Maintenance（维护）”状态的片，然后再修复处于“Last Erred（最近出错）”状态的片。

下表说明了 RAID-5 卷状态。

表 14-1 RAID-5 卷状态

状态	含义
正在初始化	片正在将所有的磁盘块都设置为零。由于 RAID-5 卷的性质与数据和奇偶校验交错分组相关，因此该过程是必需的。 状态更改为“Okay（正常）”后，初始化过程即完成，并且您可以打开设备。在此之前，应用程序会一直收到错误消息。
正常	设备可供使用且当前没有错误。

表 14-1 RAID-5 卷状态 (续)

状态	含义
维护	由于 I/O 或打开错误，已将片标记为有故障。这些错误会在读取或写入操作期间遇到。

下表说明了 RAID-5 卷的片状态以及可能执行的操作。

表 14-2 RAID-5 片状态

状态	含义	操作
正在初始化	片正在将所有的磁盘块都设置为零。由于 RAID-5 卷的性质与数据和奇偶校验交错分组相关，因此该过程是必需的。	正常，无如果在此过程期间出现 I/O 错误，则设备将进入“Maintenance（维护）”状态。如果初始化失败，则卷处于“Initialization Failed（初始化失败）”状态，而片处于“Maintenance（维护）”状态。如果发生这种情况，请清除卷并重新创建该卷。
正常	设备可供使用且当前没有错误。	无。可以添加或替换片（如有必要）。
正在重新同步	正在以活动方式重新同步片。出现错误但已更正，已启用片，或已添加片。	如果需要，请监视 RAID-5 卷状态，直到重新同步完成为止。
维护	由于 I/O 或打开错误，已将单个片标记为有故障。这些错误会在读取或写入操作期间遇到。	启用或替换出现故障的片。请参见第 174 页中的“如何在 RAID-5 卷中启用组件”或第 174 页中的“如何在 RAID-5 卷中替换组件”。 <code>metastat</code> 命令将显示 <code>invoke</code> 恢复消息，其中包含要使用 <code>metareplace</code> 命令执行的相应操作。
维护/最近出错	多个片遇到了错误。出现故障的片的状态为“Maintenance（维护）”或“Last Erred（最近出错）”。在这种情况下，不会对处于“Maintenance（维护）”状态的片尝试 I/O。但是，会对标记为“Last Erred（最近出错）”的片尝试 I/O，结果将影响 I/O 请求的总体状态。	请启用或替换出现故障的片。请参见第 174 页中的“如何在 RAID-5 卷中启用组件”或第 174 页中的“如何在 RAID-5 卷中替换组件”。 <code>metastat</code> 命令将显示 <code>invoke</code> 恢复消息，其中包含要使用 <code>metareplace</code> 命令执行的相应操作。此命令必须使用 <code>-f</code> 标志来运行。此状态表明由于存在多个有故障的片，因此可能会伪造数据。

有关替换和启用 RAID-5 卷中的片的概述

Solaris Volume Manager 能够替换和启用镜像及 RAID-5 卷中的组件。对于镜像和 RAID-5 卷，执行此操作的问题和要求是相同的。有关更多信息，请参见第 242 页中的“在 RAID-1 和 RAID-5 卷中替换和启用组件的概述”。

方案—RAID-5 卷

RAID-5 卷允许具有冗余存储，而不会带来 RAID-1 卷的额外开销费用（RAID-1 卷需要具备总存储空间的两倍容量才能提供数据冗余）。通过设置 RAID-5 卷，可以提供比使用同一组磁盘组件上的 RAID-1 卷实现的冗余存储容量更大的冗余存储。此外，借助热备件的帮助（请参见第 16 章，特别是第 180 页中的“热备件的工作原理”），可以实现几乎相同的安全级别。缺点是增加了写入时间，且在组件出现故障时明显降低了性能。但是，上述优缺点的折衷可能对于许多情况无关紧要。以下示例（利用第 5 章中说明的样例方案）介绍了 RAID-5 卷如何提供额外的存储容量。

RAID-0 和 RAID-1 卷的其他方案使用分布于 2 个控制器上的 6 张磁盘上的 6 个片（c1t1d0、c1t2d0、c1t3d0、c2t1d0、c2t2d0、c2t3d0）来提供 27 GB 的冗余存储。通过使用 RAID-5 配置中的相同的片，可以实现 45 GB 的存储。此外，该配置可以经受单个组件故障，而不会丢失数据或中断访问。通过向该配置中添加热备件，RAID-5 卷可以经受其他的组件故障。此方法的最明显的缺点是控制器故障会导致该 RAID-5 卷上的数据丢失，而第 101 页中的“方案—RAID-1 卷（镜像）”中介绍的 RAID-1 卷不会出现这种情况。

RAID-5 卷（任务）

本章提供有关执行与 RAID-5 卷相关联的 Solaris Volume Manager 任务的信息。有关这些任务中所涉及概念的信息，请参见第 14 章。

RAID-5 卷（任务列表）

以下任务列表列出了管理 Solaris Volume Manager RAID-5 卷所需的过程。

任务	说明	参考
创建 RAID-5 卷	使用 Solaris Volume Manager GUI 或 <code>metainit</code> 命令创建 RAID-5 卷。	第 170 页中的“如何创建 RAID-5 卷”
检查 RAID-5 卷的状态	使用 Solaris Volume Manager GUI 或 <code>metastat</code> 命令检查 RAID-5 卷的状态。	第 171 页中的“如何检查 RAID-5 卷的状态”
扩展 RAID-5 卷	使用 Solaris Volume Manager GUI 或 <code>metattach</code> 命令扩展 RAID-5 卷。	第 172 页中的“如何扩展 RAID-5 卷”
在 RAID-5 卷中启用片	使用 Solaris Volume Manager GUI 或 <code>metareplace</code> 命令在 RAID-5 卷中启用片。	第 174 页中的“如何在 RAID-5 卷中启用组件”
替换 RAID-5 卷中的片	使用 Solaris Volume Manager GUI 或 <code>metareplace</code> 命令替换 RAID-5 卷中的片。	第 174 页中的“如何在 RAID-5 卷中替换组件”

创建 RAID-5 卷



注意 - 如果希望运行具有 32 位内核的 Solaris 软件，或者希望使用版本低于 Solaris 9 4/03 发行版的 Solaris OS，请不要创建大于 1 TB 的卷。有关 Solaris Volume Manager 中大型卷支持的更多信息，请参见第 45 页中的“Solaris Volume Manager 中的多 TB 支持的概述”。

▼ 如何创建 RAID-5 卷

开始之前 请查看第 45 页中的“创建 Solaris Volume Manager 组件的先决条件”和第 164 页中的“创建 RAID-5 卷的背景信息”。

► 要创建 RAID-5 卷，请使用以下方法之一：

- 从 Solaris Management Console 中增强的存储工具中，打开“卷”节点。选择“操作”⇒“创建卷”，然后按照向导中的步骤操作。有关更多信息，请参见联机帮助。
- 使用以下形式的 `metainit` 命令：

```
# metainit volume-name -r component component component -i interlace-value
volume-name    指定要创建的卷的名称。
-r             指定要创建 RAID-5 卷。
component      指定要包括在 RAID-5 卷中的片或软分区。至少需要 3 个组件。
-i            指定交错值。
```

有关更多信息，请参见 `metainit(1M)` 手册页。

示例 15-1 创建由三个片组成的 RAID-5 卷

在以下示例中，RAID-5 卷 `d45` 是使用 `-r` 选项从 3 个片创建的。由于未指定交错值，因此 `d45` 使用缺省值 16 KB。系统会确认该 RAID-5 卷已设置并开始初始化该卷。

必须等待初始化完成，才能使用该 RAID-5 卷。

```
# metainit d45 -r c2t3d0s2 c3t0d0s2 c4t0d0s2

d45: RAID is setup
```

另请参见 要为文件系统准备新创建的 RAID-5 卷，请参见《系统管理指南：设备和文件系统》中的第 18 章“创建 UFS、TMPFS 和 LOFS 文件系统（任务）”。某些应用程序（如数据库）不使用文件系统，而是使用原始卷。应用程序必须能够以自己的方式来识别原始卷。

要将热备用池与 RAID-5 卷相关联，请参见第 186 页中的“如何将热备用池与卷相关联”。

维护 RAID-5 卷

▼ 如何检查 RAID-5 卷的状态

在检查 RAID-5 卷的状态时，必须对 RAID-5 卷的状态和片的状态都进行检查，才能完全了解卷的状态以及在卷的状态不是“Okay（正常）”时数据发生丢失的可能性有多大。有关详细信息，请参见第 165 页中的“有关检查 RAID-5 卷状态的概述”。

注 – RAID-5 卷的初始化或重新同步过程不能中断。

■ 要检查 RAID-5 卷的状态，请使用以下方法之一：

- 从 Solaris Management Console 中增强的存储工具中，打开“卷”节点并查看卷的状态。选择一个卷，然后选择“操作”⇒“属性”以查看更详细的信息。有关更多信息，请参见联机帮助。
- 使用 `metastat` 命令显示 RAID-5 卷的状态。

```
# metastat [-s diskset] [volume]
```

`-s diskset` 指定要对其执行该命令的磁盘集的名称。

`volume` 指定要显示的卷。

对于 RAID-5 卷中的每个片，`metastat` 命令显示以下信息：

Device 指定条带中该片的设备名称。

Start Block 指定该片的起始块。

Dbase 指定该片是否包含状态数据库副本。

State 指定该片的状态。

Hot Spare 指定该片是否将用于对出故障的片进行热备份。

示例 15-2 查看 RAID-5 卷的状态

以下示例显示对 RAID-5 卷执行 `metastat` 命令时的输出：

```
# metastat d10

d10: RAID

    State: Okay

    Interlace: 32 blocks

    Size: 10080 blocks

Original device:

    Size: 10496 blocks

    Device                Start Block  Dbase State      Hot Spare

    c0t0d0s1              330         No   Okay

    c1t2d0s1              330         No   Okay

    c2t3d0s1              330         No   Okay
```

`metastat` 命令的输出将该卷标识为 RAID-5 卷。此信息由卷名后面的 "RAID" 标志来指示。对于 RAID-5 卷中的每个片，输出中显示以下信息：

- 条带中该片的名称。
- 该片的起始块。
- 指示这些片中都不包含状态数据库副本的指示符。
- 这些片的状态。在此示例中，所有片的状态都是“Okay（正常）”。
- 该片是否作为出故障片的热备用替换片。

▼ 如何扩展 RAID-5 卷

通常，对于空间不足的 RAID-5 卷，附加组件只是一个短期的解决方案。出于性能方面的考虑，最好拥有一个“纯” RAID-5 卷。如果必须通过扩展现有的 RAID-5 卷来获得额外的存储空间，请使用此过程。



注意 – 如果希望运行具有 32 位内核的 Solaris 软件，或者希望使用版本低于 Solaris 9 4/03 发行版的 Solaris OS，请不要创建大于 1 TB 的卷。有关 Solaris Volume Manager 中多 TB 卷支持的更多信息，请参见第 45 页中的“[Solaris Volume Manager 中的多 TB 支持的概述](#)”。

开始之前 请阅读第 164 页中的“[创建 RAID-5 卷的背景信息](#)”。

- 1 请确保您具有超级用户访问权限，而且具有所有数据的最新备份。
- 2 要向 RAID-5 卷附加额外的组件，请使用以下方法之一：
 - 从 Solaris Management Console 中增强的存储工具中，打开“卷”节点，然后打开 RAID-5 卷。选择“组件”窗格，然后选择“附加组件”。请按照屏幕上的说明操作。有关更多信息，请参见联机帮助。
 - 使用以下形式的 `metattach` 命令：

```
# metattach volume-name name-of-component-to-add
volume-name           指定要扩展的 RAID-5 卷的名称。
name-of-component-to-add 指定要附加到 RAID-5 卷的组件的名称。
```

有关更多信息，请参见 `metattach(1M)` 手册页。

示例 15-3 向 RAID-5 卷中添加组件

以下示例说明如何向现有的 RAID-5 卷 `d2` 中添加片 `c2t1d0s2`。

```
# metattach d2 c2t1d0s2

d2: column is attached
```

另请参见 对于 UFS 文件系统，请针对 RAID-5 卷运行 `growfs` 命令。请参见第 41 页中的“[使用 growfs 命令扩展卷和磁盘空间](#)”。

某些应用程序（如数据库）不使用文件系统，而是使用原始卷。在这些情况下，应用程序必须能够以自己的方式来增加扩展空间。

▼ 如何在 RAID-5 卷中启用组件

如果磁盘驱动器有缺陷，则可以按照第 174 页中的“如何在 RAID-5 卷中替换组件”中的说明用系统上的另一块可用磁盘（及其片）来替换它。或者，可以修理该磁盘、为它加标号，然后运行带有 `-e` 选项的 `metareplace` 命令来重新启用该磁盘。

- 1 请确保您具有超级用户访问权限，而且具有所有数据的最新备份。
- 2 要在 RAID-5 卷中启用出故障的组件，请使用以下方法之一：
 - 从 Solaris Management Console 中增强的存储工具 中，打开“卷”节点，然后打开 RAID-5 卷。选择“组件”窗格，然后选择出故障的组件。单击“启用组件”。请按照屏幕上的说明操作。有关更多信息，请参见联机帮助。
 - 使用以下形式的 `metareplace` 命令：

```
# metareplace -e volume-name component-name
```

`-e` 指定要将出故障的组件置于可用状态并针对出故障的组件执行重新同步。

`volume-name` 指定包含出故障组件的卷的名称。

`component-name` 指定出故障组件的名称。

`metareplace` 命令会自动开始将新组件与 RAID-5 卷的其余部分重新同步。

示例 15-4 在 RAID-5 卷中启用组件

在以下示例中，RAID-5 卷 `d20` 有一个片 `c2t0d0s2`，该片具有可更正的错误。带有 `-e` 选项的 `metareplace` 命令用于启用该片。

```
# metareplace -e d20 c2t0d0s2
```

▼ 如何在 RAID-5 卷中替换组件

此任务可替换只有一个片出现故障的 RAID-5 卷中的故障片。



注意 – 如果在多个片有错误时替换某个出故障的片可能会导致数据错误。在这种情况下，将无法确保数据的完整性。

可以针对没有出现故障的设备运行 `metareplace` 命令来更改磁盘片或其他组件。此过程可用于调整 RAID-5 卷的性能。

- 1 请确保您具有超级用户访问权限，而且具有所有数据的最新备份。
- 2 使用以下方法之一来确定需要替换 RAID-5 卷的哪个片：
 - 从 Solaris Management Console 中增强的存储工具中，打开“卷”节点。然后打开 RAID-5 卷。选择“组件”窗格。查看单个组件的状态。有关更多信息，请参见联机帮助。
 - 使用 `metastat` 命令。

```
# metastat volume
```

`volume` 指定 RAID-5 卷的名称。

查找关键字短语“Needs Maintenance（需要维护）”以确定 RAID-5 卷的状态。查找关键字“Maintenance（维护）”以确定出故障的片。

- 3 使用以下方法之一来将出故障的片替换为另一个片：
 - 从 Solaris Management Console 中增强的存储工具中，打开“卷”节点。然后打开 RAID-5 卷。选择“组件”窗格。选择出故障的组件。单击“替换组件”并按照屏幕上的说明操作。有关更多信息，请参见联机帮助。
 - 使用以下形式的 `metareplace` 命令：

```
# metareplace volume-name failed-component new-component
```

- `volume-name` 是带有出故障组件的 RAID-5 卷的名称。
- `failed-component` 指定要替换的组件的名称。
- `new-component` 指定要添加到卷中、将替代出故障组件的组件。

有关更多信息，请参见 `metareplace(1M)` 手册页。

- 4 要验证替换片的状态，请使用[步骤 2](#)中介绍的方法之一。
被替换的片的状态应当为“Resyncing（正在重新同步）”或“Okay（正常）”

示例 15-5 替换 RAID-5 组件

```
# metastat d1
```

```
d1: RAID
```

```
State: Needs Maintenance
```

```
Invoke: metareplace d1 c0t14d0s6 <new device>
```

```
Interlace: 32 blocks
```

Size: 8087040 blocks

Original device:

Size: 8087520 blocks

Device	Start Block	Dbase	State	Hot Spare
c0t9d0s6	330	No	Okay	
c0t13d0s6	330	No	Okay	
c0t10d0s6	330	No	Okay	
c0t11d0s6	330	No	Okay	
c0t12d0s6	330	No	Okay	
c0t14d0s6	330	No	Maintenance	

metareplace d1 c0t14d0s6 c0t4d0s6

d1: device c0t14d0s6 is replaced with c0t4d0s6

metastat d1

d1: RAID

State: Resyncing

Resync in progress: 98% done

Interlace: 32 blocks

Size: 8087040 blocks

Original device:

Size: 8087520 blocks

Device	Start Block	Dbase	State	Hot Spare
c0t9d0s6	330	No	Okay	
c0t13d0s6	330	No	Okay	

c0t10d0s6	330	No	Okay
c0t11d0s6	330	No	Okay
c0t12d0s6	330	No	Okay
c0t4d0s6	330	No	Resyncing

在此示例中，`metastat` 命令显示 RAID-5 卷 `d1` 中出故障的片。找到可用的片后，运行 `metareplace` 命令，在命令参数中先指定出故障的片，然后指定用来替换的片。

如果没有其他的片可用，请运行带有 `-e` 选项的 `metareplace` 命令，尝试通过重新同步出故障的设备来从可能是可更正的错误中恢复。有关此过程的更多信息，请参见第 174 页中的“如何在 RAID-5 卷中启用组件”。如果存在多个错误，则必须首先替换或启用处于“Maintenance（维护）”状态的片，然后才能修复处于“Last Erred（最近出错）”状态的片。在运行 `metareplace` 命令之后，可以使用 `metastat` 命令来监视重新同步的进度。在替换过程中，卷和新片的状态都是“Resyncing（正在重新同步）”。处于这种状态的卷可以继续使用。

热备用池（概述）

本章说明 Solaris Volume Manager 如何使用热备用池。有关执行相关任务的信息，请参见第 17 章。

本章包含以下信息：

- 第 179 页中的“热备件和热备用池的概述”
- 第 182 页中的“方案—热备件”

热备件和热备用池的概述

热备用池是 Solaris Volume Manager 用于为 RAID-1（镜像）和 RAID-5 卷提供增加的数据可用性的片（热备件）的集合。在子镜像或 RAID-5 卷中出现片故障时，Solaris Volume Manager 会自动使用热备件替换出现故障的片。

注 – 热备件不适用于 RAID-0 卷或单向镜像。要使用自动替换功能，冗余数据必须是可用的。

不能使用处于空闲状态的热备件来存放数据或状态数据库副本。热备件必须保持就绪状态，以便在与其关联的卷中出现片故障时，能够立即投入使用。要使用热备件，除了系统实际要求运行的那些磁盘以外，还必须另购其他磁盘。

通过 Solaris Volume Manager，可以动态添加、删除、替换和启用热备用池中的热备件。可以使用 Solaris Management Console 或命令行实用程序来管理热备件和热备用池。有关这些任务的详细信息，请参见第 17 章。

热备件

热备件是可正常工作且可用，但未使用的片（而不是卷）。热备件会被保留，这表示它可随时替换子镜像或 RAID-5 卷中出现故障的片。

热备件用于应对出现硬件故障的情况。RAID-1 和 RAID-5 卷中的片在出现故障时将由热备件自动替换。热备件经过重新同步便可在卷中使用。修复或替换出现故障的子镜像或 RAID-5 卷片之前，可以临时使用热备件。

可以在热备用池内创建热备件。各个热备件可以属于一个或多个热备用池。例如，您可能有两个子镜像和两个热备件。可以将热备件排列成两个热备用池，并且每个群组包含两个优先级顺序不同的热备件。通过此策略，可以指定首先使用的热备件。此策略通过使更多热备件可用，还可提高可用性。

子镜像或 RAID-5 卷使用的热备件的大小，只能等于或大于子镜像或 RAID-5 卷中出现故障的片的大小。例如，如果子镜像包含 1 GB 驱动器，则该子镜像的热备件必须为 1 GB 或更大。

热备用池

热备用池是热备件的有序列表（集合）。

可以将热备件置于一个或多个热备用池中，以便利用最少的片来获得最大的灵活性和保护功能。可以将指定用作热备件的单片置于多个热备用池中，每个热备用池具有不同的片和特征。然后，可以将热备用池指定给任意数目的子镜像卷或 RAID-5 卷。

注 - 可以将单个热备用池指定给多个子镜像或 RAID-5 卷。但是，子镜像或 RAID-5 卷只能与一个热备用池关联。

热备件的工作原理

出现 I/O 错误时，Solaris Volume Manager 会根据向热备用池添加热备件的顺序，搜索热备用池中的热备件。Solaris Volume Manager 会检查热备用池，查找第一个大小等于或大于要替换的片大小的可用热备件。Solaris Volume Manager 找到的第一个足够大的热备件将用作替换磁盘。如有必要，Solaris Volume Manager 会将热备件的状态更改为“使用中”，并将自动重新同步数据。进行替换时，不会更改热备用池中的热备件的顺序。

如果是使用镜像，则会将热备件与正常运行的子镜像中的数据重新同步。如果是使用 RAID-5 卷，则会将热备件与卷中的其他片重新同步。如果在热备件列表中未找到具有足够大小的片，则出现故障的子镜像或 RAID-5 卷会进入故障状态，并且热备件保持未使用状态。如果是使用子镜像，则子镜像不会再完整地复制数据。如果是使用 RAID-5 卷，则数据冗余性不再可用。

提示 – 向热备用池添加热备件时，请按从小到大的顺序添加这些磁盘。此策略可避免使用较大热备件来替换较小的片而造成的浪费情况。

如果片遇到 I/O 错误，则出现故障的片会处于“损坏”状态。要解决这种情况，首先要修复或替换出现故障的片。然后，使用 Solaris Management Console 中增强的存储工具使片返回到“可用”状态。或者，使用 `metahs -e` 命令。

启用或替换出现故障的片之前，子镜像或 RAID-5 卷使用热备件来代替出现故障的片。随后，会在热备用池中热备件标记为“可用”。此热备件就又可以使用了。

热备用池状态

下表说明了热备用池状态和可能要执行的操作。

表 16-1 热备用池状态（命令行）

状态	含义	操作
可用	热备用池中的热备件正在运行并且可以随时接受数据。当前未在热备件中进行写入或读取。	无。
使用中	此热备用池包含的热备件正在用于替换冗余卷中出现故障的片。	诊断热备件的使用方式。然后，修复卷中正在对其使用热备件的片。
损坏	热备件或热备用池存在问题。但是，不存在立即丢失数据的危险。如果所有的热备件都正在使用，则也会显示此状态。	诊断热备件的使用方式或其损坏的原因。如果需要，可以向热备用池中添加更多热备件。

示例一热备用池

图 16-1 说明了与镜像 d1 中的子镜像 d11 和 d12 关联的热备用池。如果任一子镜像中的片出现故障，都会使用热备件自动替换出现故障的片。热备用池本身与每个子镜像卷而不是镜像关联。如果需要，热备用池还可以与其他子镜像或 RAID-5 卷关联。

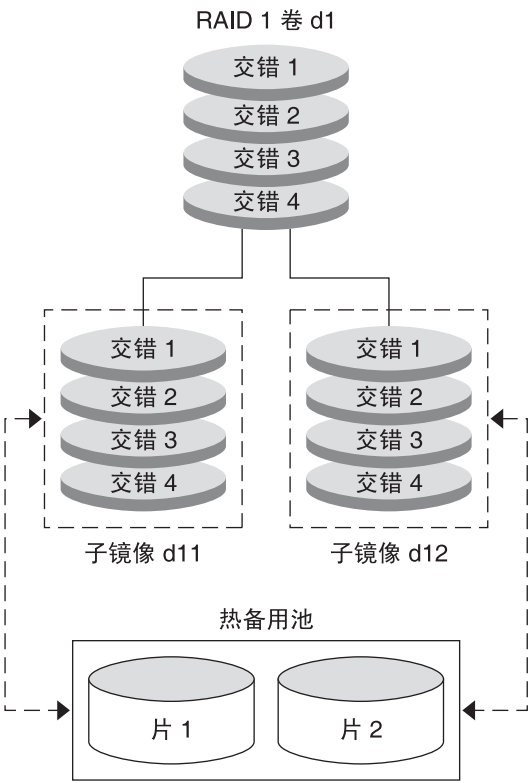


图 16-1 热备用池示例

方案—热备件

热备件可为冗余卷（RAID-1 和 RAID-5）提供额外保护，从而帮助防止数据丢失。通过将热备件与组成 RAID-0 子镜像或 RAID-5 配置的基础片关联，可以使系统自动将出现故障的片替换为热备用池中的正常片。这些交换使用的片将使用其具有的信息进行更新。然后，这些片可以像原始片一样继续运行。可以在方便的时候替换出现故障的片。

热备用池（任务）

本章介绍如何使用 Solaris Volume Manager 的热备件和热备用池。有关相关概念的信息，请参见第 16 章。

热备用池（任务列表）

以下任务列表列出了管理 Solaris OS Volume Manager 热备用池所需的过程。

任务	说明	参考
创建热备用池	使用 Solaris Volume Manager GUI 或 <code>metainit</code> 命令创建热备用池。	第 184 页中的“如何创建热备用池”
向热备用池中添加片	使用 Solaris Volume Manager GUI 或 <code>metahs</code> 命令向热备用池中添加片。	第 185 页中的“如何向热备用池中添加更多的片”
将热备用池与卷相关联	使用 Solaris Volume Manager GUI 或 <code>metaparam</code> 命令将热备用池与卷相关联。	第 186 页中的“如何将热备用池与卷相关联”
更改与卷相关联的热备用池	使用 Solaris Volume Manager GUI 或 <code>metaparam</code> 命令更改与卷相关联的热备用池。	第 188 页中的“如何更改相关的热备用池”
检查热备件和热备用池的状态	使用 Solaris Volume Manager GUI、 <code>metastat</code> 命令或 <code>metahs -i</code> 命令检查热备件或热备用池的状态。	第 190 页中的“如何检查热备件和热备用池的状态”
替换热备用池中的热备件	使用 Solaris Volume Manager GUI 或 <code>metahs</code> 命令替换热备用池中的热备件。	第 191 页中的“如何替换热备用池中的热备件”
删除热备用池中的热备件	使用 Solaris Volume Manager GUI 或 <code>metahs</code> 命令删除热备用池中的热备件。	第 192 页中的“如何从热备用池中删除热备件”

任务	说明	参考
启用热备件	使用 Solaris Volume Manager GUI 或 metahs 命令启用热备用池中的热备件。	第 193 页中的 “如何启用热备件”

创建热备用池

▼ 如何创建热备用池



注意 – 如果希望运行具有 32 位内核的 Solaris 软件，或者希望使用版本低于 Solaris 9 4/03 发行版的 Solaris OS，请不要创建大于 1 TB 的卷或热备件。有关 Solaris Volume Manager 中多 TB 卷支持的更多信息，请参见第 45 页中的 “Solaris Volume Manager 中的多 TB 支持的概述”。



注意 – 即使创建的热备件不够大，Solaris Volume Manager 也不会发出警告。如果热备件小于它所连接到的卷，则热备件将无法工作。

开始之前 请查看第 45 页中的 “创建 Solaris Volume Manager 组件的先决条件”。

- 1 成为超级用户。
- 2 要创建热备用池，请使用以下方法之一：
 - 从 Solaris Management Console 中增强的存储工具中，打开 “热备用池” 节点，然后选择 “操作” ⇒ “创建热备用池”。有关更多信息，请参见联机帮助。
 - 使用以下形式的 `metainit` 命令：

```
# metainit hot-spare-pool-name ctds-for-slice
hot-spare-pool-name    指定热备用池的名称。
ctds-for-slice         指定要添加到热备用池中的片。对要添加到热备用池中的每个片，重复此选项。
```

有关更多信息，请参见 `metainit(1M)` 手册页。

注 – `metahs` 命令还可用来创建热备用池。

示例 17-1 创建热备用池

```
# metainit hsp001 c2t2d0s2 c3t2d0s2
```

```
hsp001: Hotspare pool is setup
```

在此示例中，热备用池 `hsp001` 包含两个作为热备件的磁盘。系统会确认热备用池已设置。

另请参见 要向热备用池中添加更多的热备件，请参见第 185 页中的“如何向热备用池中添加更多的片”。在创建热备用池之后，需要将它与子镜像或 RAID-5 卷相关联。请参见第 186 页中的“如何将热备用池与卷相关联”。

▼ 如何向热备用池中添加更多的片

开始之前 请查看第 45 页中的“创建 Solaris Volume Manager 组件的先决条件”。

1 成为超级用户。

2 要向现有的热备用池中添加片，请使用以下方法之一：

- 从 Solaris Management Console 中增强的存储工具中，打开“热备用池”节点。选择要更改的热备用池。选择“操作”⇒“属性”，然后选择“热备件”面板。有关更多信息，请参见联机帮助。
- 请使用以下形式的 `metahs` 命令：

```
# metahs -a hot-spare-pool-name slice-to-add
```

```
# metahs -a -all hot-spare-pool-name slice-to-add
```

`-a hot-spare-pool-name` 指定要向指定的热备用池中添加片。

`-a all` 指定要向所有的热备用池中添加片。

`slice-to-add` 指定要添加到热备用池中的片。

有关更多信息，请参见 `metahs(1M)` 手册页。

注 可以向一个或多个热备用池中添加热备件。将热备件添加到热备用池中时，该热备件将添加到此热备用池中片列表的末尾。

示例 17-2 向一个热备用池中添加热备用片

在此示例中，`-a` 选项会将 `/dev/dsk/c3t0d0s2` 片添加到热备用池 `hsp001` 中。系统会验证该片是否已添加到热备用池中。

```
# metahs -a hsp001 /dev/dsk/c3t0d0s2
```

```
hsp001: Hotspare is added
```

示例 17-3 向所有的热备用池中添加热备用片

在此示例中，与 `all` 一起使用的 `-a` 选项将 `/dev/dsk/c3t0d0s2` 片添加到系统上配置的所有热备用池中。系统会验证该片是否已添加到所有的热备用池中。

```
# metahs -a -all /dev/dsk/c3t0d0s2
```

```
hsp001: Hotspare is added
```

```
hsp002: Hotspare is added
```

```
hsp003: Hotspare is added
```

将热备用池与卷相关联

▼ 如何将热备用池与卷相关联

开始之前 请查看第 45 页中的“创建 Solaris Volume Manager 组件的先决条件”。

- 1 成为超级用户。
- 2 要将热备用池与 RAID-5 卷或子镜像相关联，请使用以下方法之一：
 - 从 Solaris Management Console 中增强的存储工具中，打开“卷”并选择一个卷。选择“操作”⇒“属性”，然后选择“热备用池”面板。最后选择“附加 HSP”。有关更多信息，请参见联机帮助。
 - 使用以下形式的 `metaparam` 命令：

```
# metaparam -h hot-spare-pool component
```

```
-h                指定要修改指定的热备用池。
```

```
hot-spare-pool    指定热备用池的名称。
```

component 指定要将热备用池与之关联的子镜像或 RAID-5 卷的名称。

有关更多信息，请参见 `metaparam(1M)` 手册页。

示例 17-4 将热备用池与子镜像相关联

在以下示例中，`-h` 选项将热备用池 `hsp100` 与镜像 `d0` 的两个子镜像 `d10` 和 `d11` 相关联。`metastat` 命令显示热备用池与子镜像相关联。

```
# metaparam -h hsp100 d10

# metaparam -h hsp100 d11

# metastat d0

d0: Mirror

    Submirror 0: d10

        State: Okay

    Submirror 1: d11

        State: Okay

...

d10: Submirror of d0

    State: Okay

    Hot spare pool: hsp100

...

d11: Submirror of d0

    State: Okay

    Hot spare pool: hsp100
```

...

示例 17-5 将热备用池与 RAID-5 卷相关联

在以下示例中，-h 选项将热备件 hsp001 与 RAID-5 卷 d10 相关联。metastat 命令显示热备用池与 RAID-5 卷相关联。

```
# metaparam -h hsp001 d10
```

```
# metastat d10
```

```
d10: RAID
```

```
State: Okay
```

```
Hot spare pool: hsp001
```

...

▼ 如何更改相关的热备用池

开始之前 请查看第 45 页中的“创建 Solaris Volume Manager 组件的先决条件”。

- 1 成为超级用户。
- 2 要更改与某个卷相关联的热备用池，请使用以下方法之一：
 - 从 Solaris Management Console 中增强的存储工具中，打开“卷”节点并选择该卷。选择“操作”⇒“属性”，然后选择“热备用池”面板。按照屏幕上的说明，拆离不需要的热备用池，并附加新的热备用池。有关更多信息，请参见联机帮助。
 - 使以下形式的 metaparam 命令：

```
# metaparam -h hot-spare-pool-name RAID5-volume-or-submirror-name
```

-h 指定要修改指定的热备用池。

hot-spare-pool 指定新热备用池的名称，或者使用特殊关键字 none 来删除热备用池关联。

component 指定要将热备用池附加到的子镜像或 RAID-5 卷的名称。

有关更多信息，请参见 metaparam(1M) 手册页。

示例 17-6 更改热备用池的关联

在以下示例中，热备用池 `hsp001` 最初与 RAID-5 卷 `d4` 相关联。该卷的热备用池关联随后更改为 `hsp002`。`metastat` 命令显示热备用池关联在进行此更改前后的状态。

```
# metastat d4

d4: RAID

    State: Okay

    Hot spare pool: hsp001

...

# metaparam -h hsp002 d4

# metastat d4

d4: RAID

    State: Okay

    Hot spare pool: hsp002

...
```

示例 17-7 删除卷与热备用池的关联

在以下示例中，热备用池 `hsp001` 最初与 RAID-5 卷 `d4` 相关联。该热备用池关联随后更改为 `none`，这表示不应当将任何热备用池与该卷相关联。`metastat` 命令显示热备用池关联在进行此更改前后的状态。

```
# metastat d4

d4: RAID

    State: Okay

    Hot spare pool: hsp001

...

# metaparam -h none d4

# metastat d4
```

```
d4: RAID

State: Okay

Hot spare pool:

...
```

维护热备用池

以下各节说明如何针对热备用池执行维护任务。

▼ 如何检查热备件和热备用池的状态

- ▶ 要查看热备用池及其热备件的状态，请使用以下方法之一：
 - 从 Solaris Management Console 中增强的存储工具中，打开“热备用池”节点并选择一个热备用池。选择“操作”⇒“属性”以查看详细的状态信息。有关更多信息，请参见联机帮助。
 - 运行以下形式的 `metastat` 命令：

```
# metastat hot-spare-pool-name
```

示例 17-8 查看热备用池的状态

以下示例显示针对热备用池运行 `metastat` 命令时的样例输出：

```
# metastat hsp001

hsp001: 1 hot spare

c1t3d0s2          Available          16800 blocks
```

`metahs` 命令也可用来检查热备用池的状态。

有关热备用池状态以及可能执行的操作的信息，请参见第 181 页中的“热备用池状态”。

▼ 如何替换热备用池中的热备件

- 1 成为超级用户。
- 2 使用以下方法之一验证热备件当前是否正在使用：
 - 从 Solaris Management Console 中增强的存储工具中，打开“热备用池”节点并选择一个热备用池。选择“操作”⇒“属性”，然后选择“热备件”面板。请按照屏幕上的说明操作。有关更多信息，请参见联机帮助。
 - 使用以下形式的 `metastat` 命令查看热备用池的状态：

```
# metastat hot-spare-pool-name
```

有关更多信息，请参见 `metastat(1M)` 手册页。

- 3 要替换热备件，请使用以下方法之一：
 - 从 Solaris Management Console 中增强的存储工具中，打开“热备用池”节点并选择一个热备用池。选择“操作”⇒“属性”，然后选择“热备件”面板。请按照屏幕上的说明操作。有关更多信息，请参见联机帮助。
 - 使用以下形式的 `metahs` 命令：

```
# metahs -r hot-spare-pool-name current-hot-spare replacement-hot-spare
```

`-r` 指定要替换指定热备用池中的磁盘。

`hot-spare-pool-name` 指定热备用池的名称。还可以使用特殊关键字 `all` 来更改所有关联的热备用池。

`current-hot-spare` 指定将替换的当前热备件的名称。

`replacement-hot-spare` 指定将替换指定热备用池中当前热备件的片的名称。

有关更多信息，请参见 `metahs(1M)` 手册页。

示例 17-9 替换一个热备用池中的热备件

在以下示例中，`metastat` 命令显示该热备件未被使用。`metahs -r` 命令将热备用池 `hsp003` 中的热备件 `/dev/dsk/c0t2d0s2` 替换为热备件 `/dev/dsk/c3t1d0s2`。

```
# metastat hsp003
```

```
hsp003: 1 hot spare
```

```

c0t2d0s2                Broken                5600 blocks

```

```
# metahs -r hsp003 c0t2d0s2 c3t1d0s2
```

```
hsp003: Hotspare c0t2d0s2 is replaced with c3t1d0s2
```

示例 17-10 替换所有相关热备用池中的热备件

在以下示例中，关键字 `all` 将所有相关热备用池中的热备件 `/dev/dsk/clt0d0s2` 替换为热备件 `/dev/dsk/c3t1d0s2`。

```
# metahs -r all clt0d0s2 c3t1d0s2
```

```
hsp001: Hotspare clt0d0s2 is replaced with c3t1d0s2
```

```
hsp002: Hotspare clt0d0s2 is replaced with c3t1d0s2
```

```
hsp003: Hotspare clt0d0s2 is replaced with c3t1d0s2
```

▼ 如何从热备用池中删除热备件

- 1 成为超级用户。
- 2 使用以下方法之一验证热备件当前是否正在使用：
 - 从 Solaris Management Console 中增强的存储工具中，打开“热备用池”节点并选择一个热备用池。选择“操作”⇒“属性”，然后选择“热备件”面板。请按照屏幕上的说明操作。有关更多信息，请参见联机帮助。
 - 使用以下形式的 `metastat` 命令查看热备用池的状态：

```
# metastat hot-spare-pool-name
```

请参见 `metastat(1M)` 手册页。

- 3 要删除热备件，请使用以下方法之一：
 - 从 Solaris Management Console 中增强的存储工具中，打开“热备用池”节点并选择一个热备用池。选择“操作”⇒“属性”，然后选择“热备件”面板。请按照屏幕上的说明操作。有关更多信息，请参见联机帮助。
 - 使用以下形式的 `metahs` 命令：

```
# metahs -d hot-spare-pool-name current-hot-spare
```

```
-d                指定要从指定的热备用池中删除热备件。
```


hot-spare-pool 指定热备用池的名称。还可以使用特殊关键字 *all* 来将该热备件从所有的热备用池中删除。

current-hot-spare 指定将被删除的当前热备件的名称。

有关更多信息，请参见 *metahs(1M)* 手册页。

示例 17-11 从一个热备用池中删除热备件

在以下示例中，*metastat* 命令显示该热备件未被使用。*metahs -d* 命令从热备用池 *hsp003* 中删除热备件 */dev/dsk/c0t2d0s2*。

```
# metastat hsp003
```

```
hsp003: 1 hot spare
```

```
          c0t2d0s2          Broken          5600 blocks
```

```
# metahs -d hsp003 c0t2d0s2
```

▼ 如何启用热备件

- 1 成为超级用户。
- 2 要使热备件返回到“可用”状态，请使用以下方法之一：
 - 从 Solaris Management Console 中增强的存储工具中，打开“热备用池”节点并选择一个热备用池。选择“操作”⇒“属性”，然后选择“热备件”面板。请按照屏幕上的说明操作。有关更多信息，请参见联机帮助。
 - 使用以下形式的 *metahs* 命令：

```
# metahs -e hot-spare-slice
```

-e 指定要启用热备件。

hot-spare-slice 指定要启用的片的名称。

有关更多信息，请参见 *metahs(1M)* 手册页。

示例 17-12 启用热备件

在以下示例中，*metahs* 命令会在热备件 */dev/dsk/c0t0d0s2* 修复之后将其置于“可用”状态。不必指定热备用池。

```
# metahs -e c0t0d0s2
```

磁盘集（概述）

本章提供有关磁盘集的概念性信息。有关执行相关任务的信息，请参见第 19 章。

本章包括以下信息：

- 第 195 页中的“磁盘集方面的新增功能”
- 第 195 页中的“磁盘集介绍”
- 第 198 页中的“Solaris Volume Manager 磁盘集管理”
- 第 204 页中的“磁盘集使用指南”
- 第 205 页中的“磁盘集中的异步共享存储”
- 第 205 页中的“方案一磁盘集”

磁盘集方面的新增功能

本节介绍此 Solaris 发行版中新增的磁盘集功能。

有关 Solaris 新增功能的完整列表以及 Solaris 发行版的描述，请参见《Solaris 10 新增功能》。

磁盘集介绍

磁盘集是一组物理存储卷，其中包含逻辑卷和热备件。卷和热备用池必须在该磁盘集中的驱动器上生成。在磁盘集中创建卷后，可以像使用物理片一样使用卷。可以使用卷来创建和挂载文件系统，以及存储数据。

注 - 磁盘集在基于 SPARC 和 x86 的平台上均受支持。

磁盘集类型

本节介绍 Solaris Volume Manager 中可用的不同类型的磁盘集。

本地磁盘集

每台主机都具有本地磁盘集。本地磁盘集包含主机上不属于已命名磁盘集的所有磁盘。本地磁盘集专属于特定主机。本地磁盘集包含该特定主机配置的状态数据库。本地磁盘集中的卷和热备用池仅包含本地磁盘集中的驱动器。

已命名的磁盘集

除了本地磁盘集以外，主机还可以分享已命名的磁盘集。已命名的磁盘集是本地磁盘集以外的任何磁盘集。可以根据系统的配置，使用以下类型的已命名磁盘集来管理卷。

共享磁盘集

共享磁盘集可由多个主机共享。尽管共享磁盘集在所有参与的主机中都是可见的，但是只有该磁盘集的属主才能对其进行访问。每台主机都可以控制共享磁盘集，但一次只能有一台主机对其进行控制。此外，共享磁盘集还提供了用于管理卷的特定名称空间。

共享磁盘集支持数据冗余性和数据可用性。如果一台主机出现故障，另一台主机可以取代该主机的磁盘集（此类型的配置称为**故障转移配置**）。

注 - 共享磁盘集的设计目的之一是为了与 Sun Cluster、Solstice HA（High Availability，高可用性）或其他受支持的第三方 HA 框架协同工作。Solaris Volume Manager 本身不提供实现故障转移配置所必需的所有功能。

尽管每台主机都可以控制磁盘集，但一次只能有一台主机对其进行控制。

自动获取磁盘集

在 Solaris 9 4/04 发行版提供自动获取功能之前，Solaris Volume Manager 不支持通过 `/etc/vfstab` 文件在磁盘集上自动挂载文件系统。Solaris Volume Manager 要求系统管理员在访问磁盘集上的文件系统之前使用 `metaset -s setname -t` 命令手动发出磁盘集获取命令。

借助自动获取功能，可以使用 `metaset -s setname -A enable` 命令将磁盘集设置为在引导时自动访问，可以在引导时在 `/etc/vfstab` 文件中为文件系统定义挂载选项，还可以在 `/etc/vfstab` 文件中为位于已启用磁盘集的卷上的文件系统定义挂载选项。

只有单个主机磁盘集支持自动获取功能。自动获取功能要求磁盘集不得与任何其他系统共享。不能将共享的磁盘集设置为使用自动获取功能，不能对其使用 `metaset -A` 命令。但是，从磁盘集删除其他主机后，便可将该磁盘集设置为自动获取。同样，不能在自动获取磁盘集中添加其他主机。如果自动获取功能已禁用，则可以将其他主机添加到该磁盘集。

注 – 在 Sun Cluster 环境中，自动获取功能是禁用的。Sun Cluster 可以处理磁盘集的获取和释放。

有关自动获取功能的更多信息，请参见 `metaset(1M)` 命令的 `-A` 选项。

多属主磁盘集

在 Sun Cluster 环境下创建的已命名磁盘集称为多属主磁盘集。多属主磁盘集允许多个节点共享磁盘集的拥有权并同时访问共享磁盘。多属主磁盘集中的所有磁盘和卷均可由群集中的所有节点直接访问。每个多属主磁盘集都包含已添加到该磁盘集中的主机的列表。因此，群集配置中的每个多属主磁盘集可以具有一组不同（有时为重叠的）的主机。

每个多属主磁盘集都有主节点。主节点的功能是管理和更新状态数据库副本更改。由于每个磁盘集都有主节点，因此可以同时存在多个主节点。选择主节点的方法有两种。第一种方法是，如果某一节点是第一个将磁盘添加到磁盘集中的节点，则该节点将成为主节点。第二种方法是，如果主节点崩溃并出现故障，则节点 ID 最小的节点将成为主节点。

只有在 Sun Cluster 环境下才启用多属主磁盘集功能，目的是管理多属主磁盘集存储。Solaris Volume Manager for Sun Cluster 功能可与从 Sun Cluster 10/04 软件集合开始的 Sun Cluster 发行版以及 Oracle Real Applications Clusters 等应用程序配合工作。有关 Solaris Volume Manager for Sun Cluster 的更多信息，请参见第 4 章“Solaris Volume Manager for Sun Cluster（概述）”。

配置多属主磁盘集之前，除了安装 Solaris OS 以外还必须安装以下软件：

- Sun Cluster 初始群集框架
- Sun Cluster Support for Oracle Real Application Clusters 软件
- Oracle Real Application Clusters 软件

注 – 有关设置 Sun Cluster 和 Oracle Real Application Clusters 软件的信息，请参见《Sun Cluster 软件安装指南（适用于 Solaris OS）》和《用于 Oracle Real Application Clusters 的 Sun Cluster 数据服务指南（适用于 Solaris OS）》。

Solaris Volume Manager 磁盘集管理

与本地磁盘集管理不同的是，您无需手动创建或删除磁盘集状态数据库。Solaris Volume Manager 会在磁盘集中的每个磁盘上放置一个状态数据库副本（在片 7 上），最多可在磁盘集中放置 50 个副本。

向磁盘集中添加磁盘时，Solaris Volume Manager 会在磁盘集上自动创建状态数据库副本。将磁盘收入磁盘集时，Solaris Volume Manager 可能会对磁盘重新分区，以便可将磁盘集的状态数据库副本置于磁盘上（请参见第 200 页中的“自动磁盘分区”）。

驻留在磁盘集中的卷上的文件系统通常不会在引导时通过 `/etc/vfstab` 文件自动挂载。必要的 Solaris Volume Manager RPC 守护进程（`rpc.metad` 和 `rpc.metamhd`）不会在引导过程中很早就启动以允许此操作发生。此外，在重新引导期间会丢失磁盘集的拥有权。请不要在 `/etc/inetd.conf` 文件中禁用 Solaris Volume Manager RPC 守护进程。它们被配置为在缺省情况下启动。只有这些守护进程始终处于启用状态，Solaris Volume Manager 才能使用其全部功能。

如果使用 `metaset` 命令的 `-A` 选项启用了自动获取功能，则在引导时会自动获取磁盘集。在这样的情况下，驻留在磁盘集中的卷上的文件系统可以随 `/etc/vfstab` 文件一同自动挂载。要在引导过程中启用自动获取功能，必须仅将磁盘集与单个主机关联，且必须启用自动获取功能。可以在磁盘集创建期间或创建之后启用该磁盘集。有关自动获取功能的更多信息，请参见第 196 页中的“自动获取磁盘集”。

注—尽管磁盘集在单个主机配置中受支持，但它们通常不适合在“本地”（并非双连接）使用。两种常见的例外情况是：使用磁盘集为逻辑卷提供一个更便于管理的名称空间，以及使用磁盘集更方便地管理存储区域网络 (Storage Area Network, SAN) 光纤网络上的存储器（请参见第 205 页中的“方案—磁盘集”）。

可以使用 Solaris Volume Manager 命令行界面（`metaset` 命令）或 Solaris Management Console 中增强的存储工具来创建和配置磁盘集。

向磁盘集添加磁盘后，该磁盘集便可由磁盘集中的主机**保留**（或**获取**）和**释放**。如果某一主机保留了磁盘集，则磁盘集中的其他主机便不能访问该磁盘集中磁盘上的数据。要对磁盘集执行维护，主机必须是磁盘集的属主，或已保留了磁盘集。主机通过将第一个磁盘置于磁盘集中获得该磁盘集的隐式拥有权。

可以使用 `metaimport` 命令将磁盘集（包括在不同系统上创建的磁盘集）导入现有的 Solaris Volume Manager 配置中。

保留磁盘集

主机使用磁盘集中的磁盘之前，必须先保留该磁盘集。保留磁盘集的方法有两种：

- **安全**—安全保留磁盘集时，Solaris Volume Manager 将尝试获取磁盘集，而其他主机将尝试释放磁盘集。释放可能会失败，因此保留也可能会失败。

- **强制**—强制保留磁盘集时，无论其他主机当前是否已保留磁盘集，Solaris Volume Manager 都将保留磁盘集。当磁盘集中的某一主机停机或当前没有通信时，通常会使用此方法。磁盘集中的所有磁盘都将被接管。将在执行保留的主机上读取状态数据库，并且在磁盘集中配置的共享卷变为可访问状态。如果此时其他主机保留了磁盘集，则该主机可能会由于保留丢失而崩溃。

正常情况下，磁盘集中的两个主机彼此协同工作，以便确保磁盘集中的磁盘一次只由一台主机保留。正常情况是指两个主机都在运行且彼此进行通信。

注—如果突然确定不保留磁盘（或许由于使用该磁盘集的其他主机强制获取了磁盘），则主机将崩溃。此行为有助于最大程度地减少在两台主机同时访问同一磁盘时造成的数据丢失。

有关获取或保留磁盘集的更多信息，请参见第 217 页中的“如何提取磁盘集”。

释放磁盘集

对磁盘集中的物理磁盘执行维护时，释放磁盘集可能非常有用。释放磁盘集后，主机便不能访问该磁盘集。如果磁盘集中的两台主机都释放了磁盘集，则磁盘集中的两台主机都不能访问该磁盘集中的磁盘。

有关释放磁盘集的更多信息，请参见第 219 页中的“如何释放磁盘集”。

导入磁盘集

从 Solaris 9 9/04 发行版开始，使用 `metainport` 命令可以将磁盘集（包括复制的磁盘集）导入在磁盘集中具备设备 ID 支持的现有 Solaris Volume Manager 配置。还可以使用 `metainport` 命令报告可以导入的磁盘集。

复制的磁盘集是使用远程复制软件创建的。为了使用 `metainport` 命令导入复制的磁盘集，还必须将包含磁盘集中每个磁盘的状态数据库副本的片复制到被复制磁盘集所在的同一片上。对于非 EFI 磁盘，这对应于片 7，对于 EFI 磁盘，则对应于片 6。复制磁盘集之前，请确保要复制的数据磁盘配置与远程站点的磁盘配置匹配。此步骤可确保精确复制状态数据库副本和数据。

如果磁盘不包含卷或状态数据库副本，则 `metainport` 命令也不会将磁盘集导入磁盘。如果卷或状态数据库副本尚未添加到磁盘或已从磁盘删除，则会出现此情况。在这种情况下，如果将磁盘集导入其他系统，则会发现磁盘集中缺少该磁盘。例如，每个 Solaris Volume Manager 磁盘集最多允许 50 个状态数据库副本。如果磁盘集中具有 60 个磁盘，则不包含状态数据库副本的 10 个磁盘必须包含卷，才能随磁盘集一同导入。

有关与导入磁盘集关联的任务，请参见第 222 页中的“导入磁盘集”。

注 – 在 Sun Cluster 环境下，`metainport` 命令不可用。

自动磁盘分区

向磁盘集中添加新磁盘时，Solaris Volume Manager 将检查磁盘格式。如有必要，Solaris Volume Manager 将对磁盘重新分区，以确保磁盘具有正确配置的片 7（具有容纳状态数据库副本所需的足够磁盘空间）。片 7 的准确大小取决于磁盘的几何参数。但是，大小不会小于 4 MB，可能更接近于 6 MB（具体取决于柱面边界所在的位置）。

缺省情况下，Solaris Volume Manager 会将一个状态数据库副本置于片 7 上。可以增加片 7 的缺省大小或缩小状态数据库副本的大小，以便将多个状态数据库副本置于该片中。

注 – 基于各种因素（包括状态数据库副本的大小以及要在状态数据库副本中存储的信息），片 7 的最小大小可能会在将来发生更改。多属主磁盘集中的状态数据库副本的缺省大小为 16 MB。

为了在磁盘集中使用，磁盘的片 7 必须满足以下条件：

- 在扇区 0 处开始
- 包括足够的空间，可容纳磁盘标号和状态数据库副本
- 不能挂载
- 不与任何其他片重叠，包括片 2

如果现有分区表不满足这些条件，则 Solaris Volume Manager 将对磁盘重新分区。每个驱动器都有一小部分空间保留在片 7 中供 Solaris Volume Manager 使用。每个驱动器上其余的空间都被置于片 0 中。磁盘上的任何现有数据都会因重新分区而丢失。

提示 – 向磁盘集添加驱动器后，可以在必要时对其重新分区，但不能以任何方式对片 7 进行更改。

`prtvtoc` 命令的以下输出显示了某磁盘在添加到磁盘集之前的状态。

```
[root@lexicon:apps]$ prtvtoc /dev/rdisk/c1t6d0s0

* /dev/rdisk/c1t6d0s0 partition map

*

* Dimensions:
```



```
*      512 bytes/sector

*      133 sectors/track

*      27 tracks/cylinder

*      3591 sectors/cylinder

*      4926 cylinders

*      4924 accessible cylinders

*

* Flags:

*      1: unmountable

*     10: read-only

*

*                               First      Sector      Last

* Partition  Tag  Flags      Sector      Count      Sector  Mount Directory

      0      2    00          0    4111695    4111694

      1      3    01    4111695    1235304    5346998

      2      5    01          0  17682084  17682083

      3      0    00    5346999    4197879    9544877

      4      0    00    9544878    4197879   13742756

      5      0    00   13742757    3939327   17682083
```

以上输出显示磁盘不包含片 7。因此，向磁盘集添加磁盘时，Solaris Volume Manager 将对磁盘重新分区。prtvtoc 命令的以下输出显示了该磁盘在添加到磁盘集之后的状态。

```
[root@lexicon:apps]$ prtvtoc /dev/rdisk/clt6d0s0

* /dev/rdisk/clt6d0s0 partition map

*

* Dimensions:
```

```
*      512 bytes/sector

*      133 sectors/track

*      27 tracks/cylinder

*      3591 sectors/cylinder

*      4926 cylinders

*      4924 accessible cylinders

*

* Flags:

*      1: unmountable

*     10: read-only

*

*                               First      Sector      Last

* Partition  Tag  Flags      Sector      Count      Sector  Mount Directory

      0      0    00      10773  17671311  17682083

      7      0    01          0    10773    10772

[root@lexicon:apps]$
```

输出显示磁盘已重新分区，其中包括一个在柱面 0 处开始且有足够空间容纳状态数据库副本的片 7。如果向磁盘集添加的磁盘具有可接受的片 7，则不会对它们重新格式化。

注 – 如果已使用 Solstice DiskSuite 软件升级了磁盘集，则这些磁盘集上的缺省状态数据库副本大小将为 1034 个块，而不是 Solaris Volume Manager 中的 8192 块大小。相应地，在 Solstice DiskSuite 软件下添加的磁盘上的片 7 比 Solaris Volume Manager 下添加的磁盘上的片 7 也要小。

磁盘集名称要求

磁盘集卷名称与其他 Solaris Volume Manager 组件名称类似。但是，磁盘集名称会作为该名称的一部分。例如，卷路径名称会在 `/dev/md/` 之后和路径中实际卷名称之前包含磁盘集名称。

下表显示了一些示例磁盘集卷名称。

表 18-1 磁盘集的示例卷名称

<code>/dev/md/blue/dsk/d0</code>	磁盘集 <code>blue</code> 中的块卷 <code>d0</code>
<code>/dev/md/blue/dsk/d1</code>	磁盘集 <code>blue</code> 中的块卷 <code>d1</code>
<code>/dev/md/blue/rdisk/d126</code>	磁盘集 <code>blue</code> 中的原始卷 <code>d126</code>
<code>/dev/md/blue/rdisk/d127</code>	磁盘集 <code>blue</code> 中的原始卷 <code>d127</code>

同样，热备用池也将磁盘集名称作为热备件名称的一部分。

示例—两个共享磁盘集

图 18-1 显示使用两个磁盘集的示例配置。

在此配置中，主机 A 和主机 B 共享磁盘集 `red` 和 `blue`。每个磁盘集都有各自的本地磁盘集，本地磁盘集不是共享的。如果主机 A 出现了故障，则主机 B 可以取代主机 A 来控制其共享磁盘集，即磁盘集 `red`。同样，如果主机 B 出现故障，则主机 A 可以取代主机 B 来控制其共享磁盘集，即磁盘集 `blue`。

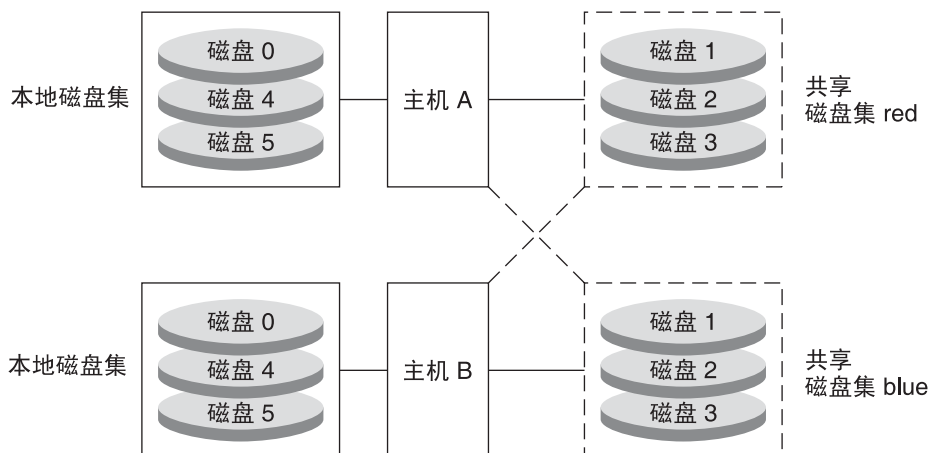


图 18-1 磁盘集示例

磁盘集使用指南

使用磁盘集时，请考虑以下指南：

- 必须在与磁盘集连接的每台主机上都配置 Solaris Volume Manager。
- 创建磁盘集之前，每台主机都必须已设置其本地状态数据库。
- 创建磁盘集和为磁盘集创建卷的一系列步骤为：首先创建磁盘集；然后，向磁盘集添加磁盘；最后，在磁盘集中创建卷。
- 要在群集环境中创建和使用磁盘集，root 必须为组 14 的成员。或者，每台主机上的 `/etc/rhosts` 文件都必须包含与该磁盘集关联的其他主机名的对应项。

注 - 此步骤在 SunCluster 3.x 环境中不是必需的。

- 要对磁盘集执行维护，主机必须是磁盘集的属主，或已保留了磁盘集。主机通过将第一个驱动器置于磁盘集中获得该磁盘集的隐式拥有权。
- 不能向磁盘集中添加当前用于文件系统、数据库或任何其他应用程序的驱动器。添加驱动器之前，请确保该驱动器当前没有被使用。
- 不要向磁盘集中添加包含要保留的现有数据的驱动器。向磁盘集添加磁盘的过程中，将对磁盘重新分区并销毁现有数据。
- 与本地卷管理不同的是，不必在磁盘集上手动创建或删除状态数据库副本。Solaris Volume Manager 将尝试在磁盘集的所有驱动器中平衡状态数据库副本的合理数目。
- 向磁盘集中添加驱动器时，Solaris Volume Manager 将重新平衡其余驱动器中的状态数据库副本。以后，如有必要，可以使用 `metadb` 命令来更改副本布局。

磁盘集中的异步共享存储

在以前的 Solaris Volume Manager 版本中，计划在磁盘集中的主机之间共享的所有磁盘都必须与每台主机相连。它们还必须在每台主机上都具有完全相同的路径、驱动程序和名称。具体来说，共享磁盘驱动器必须由位于同一位置 (/dev/rdisk/c#t#d#) 的两台主机看到。此外，共享磁盘必须使用相同的驱动程序名称 (ssd)。

在最新的 Solaris OS 发行版中，对通常可访问存储具有不同视图的系统可以采用非并行方式共享对磁盘集的访问权限。由于引入了对磁盘集的设备 ID 支持，因此 Solaris Volume Manager 可以自动跟踪已命名磁盘集中的磁盘动作。

注 – 对磁盘集的设备 ID 支持在 Sun Cluster 环境中不受支持。

升级到最新 Solaris OS 时，需要立即获取磁盘集以便启用磁盘跟踪。有关获取磁盘集的更多信息，请参见第 217 页中的“如何提取磁盘集”。

如果自动获取功能未启用，则必须手动获取每个磁盘集。如果此功能已启用，则重新引导系统时会自动执行此步骤。有关自动获取功能的更多信息，请参见第 196 页中的“自动获取磁盘集”。

通过该扩展设备 ID 支持，您还可以导入各种磁盘集（即使是在不同的系统中创建的）。有关导入磁盘集的更多信息，请参见第 199 页中的“导入磁盘集”。

方案—磁盘集

以下示例（利用第 5 章中显示的样例系统）介绍应如何使用磁盘集来管理驻留在 SAN（Storage Area Network，存储区域网络）光纤网络上的存储器。

假设样例系统具有与光纤网络交换机和 SAN 存储器连接的其他控制器。与其他设备（如 SCSI 和 IDE 磁盘）一样，SAN 光纤网络上的存储器在引导过程的初期便不能用于系统。此外，Solaris Volume Manager 还将报告光纤网络上的逻辑卷在引导时不可用。但是，通过向磁盘集添加存储器，然后使用磁盘集工具管理该存储器，可以避免引导时可用性的问题。另外，还可以在本地存储器中的单独的、由磁盘集控制的名称空间中方便地管理与光纤网络连接的存储器。

磁盘集（任务）

本章提供有关执行磁盘集相关任务的信息。有关这些任务中所涉及概念的信息，请参见第 18 章。

磁盘集（任务列表）

以下任务列表列出了管理 Solaris Volume Manager 磁盘集和 Solaris Volume Manager for Sun Cluster 多属主磁盘集所需的过程。除非另行说明，否则所有的命令都适用于这两种类型的磁盘集。Solaris Volume Manager GUI 不适用于与多属主磁盘集相关联的任务。

任务	说明	参考
创建磁盘集	使用 Solaris Volume Manager GUI 或 <code>metaset</code> 命令创建磁盘集。 使用 <code>metaset -M</code> 命令创建多属主磁盘集。	第 208 页中的 “如何创建磁盘集”
向磁盘集内添加磁盘	使用 Solaris Volume Manager GUI 或 <code>metaset</code> 命令向磁盘集内添加磁盘。	第 210 页中的 “如何向磁盘集内添加磁盘”
向磁盘集内添加主机	使用 Solaris Volume Manager GUI 或 <code>metaset</code> 命令向磁盘集内添加主机。	第 211 页中的 “如何向磁盘集内添加另一台主机”
在磁盘集内创建 Solaris Volume Manager 卷	使用 Solaris Volume Manager GUI 或 <code>metainit</code> 命令在磁盘集内创建卷。	第 212 页中的 “如何在磁盘集内创建 Solaris Volume Manager 组件”
检查磁盘集的状态	使用 Solaris Volume Manager GUI 或 <code>metaset</code> 命令检查磁盘集的状态。	第 215 页中的 “如何检查磁盘集的状态”
从磁盘集内删除磁盘	使用 Solaris Volume Manager GUI 或 <code>metaset</code> 命令从磁盘集内删除磁盘。	第 216 页中的 “如何从磁盘集内删除磁盘”

任务	说明	参考
提取磁盘集	使用 Solaris Volume Manager GUI 或 <code>metaset</code> 命令提取磁盘集。	第 217 页中的 “如何提取磁盘集”
释放磁盘集	使用 Solaris Volume Manager GUI 或 <code>metaset</code> 命令释放磁盘集。	第 219 页中的 “如何释放磁盘集”
从磁盘集内删除主机	使用 Solaris Volume Manager GUI 或 <code>metaset</code> 命令从磁盘集内删除主机。	第 220 页中的 “如何删除主机或磁盘集”
删除磁盘集	使用 Solaris Volume Manager GUI 或 <code>metaset</code> 命令删除磁盘集内的最后一台主机，从而删除磁盘集。	第 220 页中的 “如何删除主机或磁盘集”
导入磁盘集	使用 <code>metainport</code> 命令，针对磁盘集运行报告以确定哪些磁盘集可以导入，并将磁盘集从一个系统导入到另一个系统。	第 222 页中的 “导入磁盘集”

创建磁盘集

▼ 如何创建磁盘集

开始之前 请查看[第 204 页中的 “磁盘集使用指南”](#)。

1 要创建磁盘集，请使用以下方法之一：

- 从 Solaris Management Console 中增强的存储工具中，打开“磁盘集”节点。选择“操作”⇒“创建磁盘集”，然后按照向导中的说明操作。有关更多信息，请参见联机帮助。
- 要使用命令行从头创建磁盘集，请使用以下形式的 `metaset` 命令：

```
# metaset -s diskset-name -a -h -M hostname
```

`-s diskset-name` 指定执行 `metaset` 命令所针对的磁盘集的名称。

`-a` 向指定的磁盘集内添加主机。Solaris Volume Manager 支持每个磁盘集接受多达四台主机。

`-M` 指定所创建的磁盘集是多属主磁盘集。

`-h hostname` 指定一个或多个要添加到磁盘集内的主机。添加第一台主机会创建磁盘集。随后可以添加第二台主机。但是，如果磁盘集内的所有磁盘均无法在指定的 `hostname` 上找到，则不接受第二台主机。`hostname` 与在 `/etc/nodename` 文件中找到的名称相同。

有关更多信息，请参见 `metaset(1M)` 手册页。

2 检查新磁盘集的状态。

```
# metaset
```

示例 19-1 创建磁盘集

在以下示例中，将从主机 `host1` 创建一个名为 `blue` 的共享磁盘集。`metaset` 命令用来显示该磁盘集的状态。此时，磁盘集没有属主。缺省情况下，向磁盘集内添加磁盘的主机将变成属主。

```
# metaset -s blue -a -h host1
```

```
# metaset
```

```
Set name = blue, Set number = 1
```

Host	Owner
------	-------

host1	
-------	--

示例 19-2 创建多属主磁盘集

以下示例创建一个名为 `red` 的多属主磁盘集。`metaset` 命令的第一行输出显示 "Multi-owner"，这表明磁盘集是多属主磁盘集。

```
# metaset -s red -a -M -h nodeone
```

```
# metaset -s red
```

```
Multi-owner Set name = red, Set number = 1, Master =
```

Host	Owner	Member
------	-------	--------

nodeone		Yes
---------	--	-----

扩展磁盘集

▼ 如何向磁盘集内添加磁盘



注意 - 如果希望运行具有 32 位内核的 Solaris 软件，或者希望使用版本低于 Solaris 9 4/03 发行版的 Solaris OS，请不要向磁盘集内添加大于 1 TB 的磁盘。有关 Solaris Volume Manager 中多 TB 卷支持的更多信息，请参见第 45 页中的“Solaris Volume Manager 中的多 TB 支持的概述”。

只有符合以下条件的磁盘才能添加到磁盘集中：

- 磁盘不得在卷或热备用池中使用。
- 磁盘中不得包含状态数据库副本。
- 磁盘当前不得已挂载、处于交换状态，或者处于打开状态以供某应用程序使用。

开始之前 请查看第 204 页中的“磁盘集使用指南”。

1 要向磁盘集内添加磁盘，请使用以下方法之一：

- 从 Solaris Management Console 中增强的存储工具中，打开“磁盘集”节点。选择要修改的磁盘集。然后单击鼠标右键并选择“属性”。选择“磁盘”选项卡。单击“添加磁盘”，然后按照向导中的说明操作。有关更多信息，请参见联机帮助。
- 要使用命令行向磁盘集内添加磁盘，请使用以下形式的 `metaset` 命令：

```
# metaset -s diskset-name -a disk-name
```

`-s diskset-name` 指定执行 `metaset` 命令所针对的磁盘集的名称。

`-a` 向指定的磁盘集内添加磁盘。

`disk-name` 指定要添加到磁盘集内的磁盘。磁盘的名称采用以下格式：
: cxtxdx。在向磁盘集内添加磁盘时，不包括 "sx" 片标识符。

有关更多信息，请参见 `metaset(1M)` 手册页。

向磁盘集内添加磁盘的第一台主机会变成磁盘集的属主。



注意 - 请不要向磁盘集内添加包含数据的磁盘。在向磁盘集内添加包含数据的磁盘的过程中，可能会对磁盘重新分区，从而销毁数据。

2 检验磁盘集和磁盘的状态。

```
# metaset
```

示例 19-3 向磁盘集内添加磁盘

```
# metaset -s blue -a c1t6d0
```

```
# metaset
```

```
Set name = blue, Set number = 1
```

Host	Owner
host1	Yes

Drive	Dbase
c1t6d0	Yes

在此示例中，主机名是 `host1`。共享的磁盘集是 `blue`。只有磁盘 `c1t6d0` 添加到了磁盘集 `blue` 中。

或者，也可以通过在命令行上列出各个磁盘来同时添加多个磁盘。例如，可以使用以下命令同时向磁盘集内添加两个磁盘：

```
# metaset -s blue -a c1t6d0 c2t6d0
```

▼ 如何向磁盘集内添加另一台主机

以下过程说明如何向只有一台主机的现有磁盘集内添加另一台主机。Solaris Volume Manager 支持每个磁盘集接受多达四台的主机。

开始之前 请查看第 204 页中的“磁盘集使用指南”。

1 要向磁盘集内添加主机，请使用以下方法之一：

- 从 Solaris Management Console 中增强的存储工具中，打开“磁盘集”节点，然后选择要修改的磁盘集。然后单击鼠标右键并选择“属性”。选择“主机”选项卡。单击“添加主机”，然后按照向导中的说明操作。有关更多信息，请参见联机帮助。
- 要使用命令行向磁盘集内添加主机，请使用以下形式的 `metaset` 命令：

```
# metaset -s diskset-name -a -h hostname

-s diskset-name    指定要向其中添加主机的磁盘集的名称。

-a                向指定的磁盘集内添加主机。

-h hostname        指定一个或多个要添加到磁盘集内的主机名。添加第一台主机会
                  创建磁盘集。主机名与在 /etc/nodename 文件中找到的名称相
                  同。
```

有关更多信息，请参见 `metaset(1M)` 手册页。

2 验证该主机是否已添加到磁盘集内。

```
# metaset
```

示例 19-4 向磁盘集内添加另一台主机

```
# metaset -s blue -a -h host2

# metaset

Set name = blue, Set number = 1
```

Host	Owner
host1	Yes
host2	

Drive	Dbase
c1t6d0	Yes
c2t6d0	Yes

此示例说明了如何将主机 `host2` 添加到磁盘集 `blue` 中。

▼ 如何在磁盘集内创建 Solaris Volume Manager 组件

在创建磁盘集之后，可以使用已添加到磁盘集内的磁盘创建卷和热备用池。可以使用 Solaris Management Console 中增强的存储工具或命令行实用程序。

- 要在磁盘集内创建卷或其他 Solaris Volume Manager 组件，请使用以下方法之一：

 - 从 Solaris Management Console 中增强的存储工具中，打开“卷”、“状态数据库副本”或“热备用池”节点。选择“操作”⇒“创建”，然后按照向导中的说明操作。有关更多信息，请参见联机帮助。
 - 使用具有相同基本语法的相同命令行实用程序来创建卷、状态数据库副本或热备用池。但是，在每个命令后面都还要紧跟一个 `-s disk-set`。

```
# command -s disk-set
```

示例 19-5 在磁盘集内创建 Solaris Volume Manager 卷

以下示例说明如何在磁盘集 `blue` 内创建镜像 `d10`。该镜像由子镜像（RAID-0 卷）`d11` 和 `d12` 组成。

```
# metainit -s blue d11 1 1 c1t6d0s0

blue/d11: Concat/Stripe is setup

# metainit -s blue d12 1 1 c2t6d0s0

blue/d12: Concat/Stripe is setup

# metainit -s blue d10 -m d11

blue/d10: Mirror is setup

# metattach -s blue d10 d12

blue/d10: submirror blue/d12 is attached


# metastat -s blue

blue/d10: Mirror

    Submirror 0: blue/d11

        State: Okay

    Submirror 1: blue/d12

        State: Resyncing

    Resync in progress: 0 % done
```

Pass: 1

Read option: roundrobin (default)

Write option: parallel (default)

Size: 17674902 blocks

blue/d11: Submirror of blue/d10

State: Okay

Size: 17674902 blocks

Stripe 0:

Device	Start Block	Dbase	State	Reloc	Hot Spare
c1t6d0s0	0	No	Okay		

blue/d12: Submirror of blue/d10

State: Resyncing

Size: 17674902 blocks

Stripe 0:

Device	Start Block	Dbase	State	Reloc	Hot Spare
c2t6d0s0	0	No	Okay		

维护磁盘集

▼ 如何检查磁盘集的状态

- ▶ 可使用以下方法之一检查磁盘集的状态：
 - 从 Solaris Management Console 中增强的存储工具中，打开“磁盘集”节点。在要监视的磁盘集上单击鼠标右键。然后从弹出的菜单中选择“属性”。有关更多信息，请参见联机帮助。
 - 使用以下形式的 `metaset` 命令查看磁盘集的状态。

```
# metaset -s diskset-name
```

有关更多信息，请参见 `metaset(1M)` 手册页。

注 - 磁盘集拥有权仅显示在所属主机上。

示例 19-6 检查指定磁盘集的状态

以下示例说明带有 `-s` 选项的 `metaset` 命令，后跟磁盘集名称 `blue`。此命令的输出中显示该磁盘集的状态信息。此命令的输出表明 `host1` 是磁盘集的属主。`metaset` 命令还显示磁盘集内的磁盘。

```
red# metaset -s blue
```

```
Set name = blue, Set number = 1
```

Host	Owner
host1	Yes
Drive	Dbase
c1t6d0	Yes

c2t6d0 Yes

如果只使用 `metaset` 命令，则显示所有磁盘集的状态。

▼ 如何从磁盘集内删除磁盘

1 要从磁盘集内删除磁盘，请使用以下方法之一：

- 从 Solaris Management Console 中增强的存储工具中，打开“磁盘集”节点。在要释放的磁盘集上单击鼠标右键。然后从弹出的菜单中选择“属性”。单击“磁盘”选项卡。按照联机帮助中的说明操作。
- 使用以下形式的 `metaset` 命令从磁盘集内删除磁盘：

```
# metaset -s diskset-name -d disk-name
```

-s *diskset-name* 指定要删除磁盘的磁盘集的名称。

-d *disk-name* 指定要从磁盘集内删除的磁盘。磁盘的名称采用以下格式：
 : cxtxdx。在从磁盘集内删除磁盘时，不包括 "sx" 片标识符。

有关更多信息，请参见 `metaset(1M)` 手册页。

2 验证该磁盘已从磁盘集内删除。

```
# metaset -s diskset-name
```

注 – 要删除磁盘集，必须首先删除其中的所有磁盘。

示例 19-7 从磁盘集内删除磁盘

以下示例说明如何从磁盘集 `blue` 内删除磁盘 `c1t6d0`：

```
host1# metaset -s blue -d c1t6d0
```

```
host1# metaset -s blue
```

```
Set name = blue, Set number = 1
```

Host	Owner
------	-------

host1	
host2	
Drive	Dbase
c2t6d0	Yes

▼ 如何提取磁盘集

注 - 此选项对于多属主磁盘集不可用。

■ 使用以下方法之一提取磁盘集：

- 从 Solaris Management Console 中增强的存储工具中，打开“磁盘集”节点。在要提取的磁盘集上单击鼠标右键。然后从弹出的菜单中选择“获取拥有权”。有关更多信息，请参见联机帮助。
- 使用以下形式的 `metaset` 命令：

```
# metaset -s diskset-name -t -f
```

-s *diskset-name* 指定要提取的磁盘集的名称。

-t 指定要提取磁盘集。

-f 指定要强制提取磁盘集。

有关更多信息，请参见 `metaset(1M)` 手册页。

磁盘集在一段时间内只能由一台主机拥有。当某个磁盘集内的一台主机提取该磁盘集后，该磁盘集内的其他主机就不能访问该磁盘集内磁盘上的数据。

`metaset` 命令的缺省行为是：只有拥有该磁盘集的主机可以释放磁盘集时，才允许其他主机提取磁盘集。使用 `-f` 选项可强制提取磁盘集。无论该磁盘集当前是否为另一台主机所有，此选项都会提取该磁盘集。当磁盘集内的一台主机已关闭或者无法与之通信时，可以使用此方法。如果此时另一台主机已经提取了该磁盘集，那么，当该主机尝试针对此磁盘集执行 I/O 操作时，将会崩溃。

注 - 磁盘集拥有权仅显示在所属主机上。

示例 19-8 提取磁盘集

在以下示例中，主机 `host1` 与主机 `host2` 通信。此通信可确保在主机 `host1` 尝试提取该磁盘集之前，主机 `host2` 已释放该磁盘集。

```
host1# metaset
...
Set name = blue, Set number = 1
```

Host	Owner
host1	
host2	
...	

```
host1# metaset -s blue -t
```

```
host2# metaset
...
Set name = blue, Set number = 1
```

Host	Owner
host1	Yes
host2	
...	

如果 `host2` 拥有磁盘集 `blue`，则上面的输出中 "Owner" 列仍将保持空白。`metaset` 命令仅显示发出该命令的主机是否拥有该磁盘集。

示例 19-9 强制提取磁盘集

在以下示例中，正在提取该磁盘集的主机无法与另一台主机通信。`-f` 选项可允许在不发出警告的情况下强制提取该磁盘集内的磁盘。如果另一台主机拥有该磁盘集，那么，当该主机尝试针对此磁盘集执行 I/O 操作时，该主机将会崩溃。

```
# metaset -s blue -t -f
```

▼ 如何释放磁盘集

针对磁盘集内的物理磁盘执行维护时，释放磁盘集会非常有用。释放磁盘集后，主机便不能访问该磁盘集。如果磁盘集内有两台主机，且这两台主机同时释放某磁盘集，则这两台主机都无法直接访问在该磁盘集内定义的卷或热备用池。但是，在两台主机都释放某磁盘集时，这两台主机都可以通过磁盘的 `c*t*d*` 名称直接访问磁盘。

注- 此选项对于多属主磁盘集不可用。

开始之前 请查看第 204 页中的“磁盘集使用指南”。

1 使用以下方法之一释放磁盘集：

- 从 Solaris Management Console 中增强的存储工具中，打开“磁盘集”节点。在要释放的磁盘集上单击鼠标右键。然后从弹出的菜单中选择“释放拥有权”。有关更多信息，请参见联机帮助。
- 要释放磁盘集的拥有权，请使用以下形式的 `metaset` 命令：

```
# metaset -s diskset-name -r
```

`-s diskset-name` 指定执行 `metaset` 命令所针对磁盘集的名称。

`-r` 释放磁盘集的拥有权。将取消对磁盘集内所有磁盘的保留。磁盘集内的卷将不再可访问。

有关更多信息，请参见 `metaset(1M)` 手册页。

注- 磁盘集拥有权仅显示在所属主机上。

2 验证该磁盘集是否已在此主机上释放。

```
# metaset
```

示例 19-10 释放磁盘集

以下示例说明如何释放磁盘集 `blue`。请注意，该磁盘集没有属主。从主机 `host1` 查看状态可能会引起误导。主机只能确定它是否拥有磁盘集。例如，如果主机 `host2` 将要获取该磁盘集的拥有权，那么，该拥有权将不会出现在主机 `host1` 上。只有主机 `host2` 才将显示 `host2` 对该磁盘集具有拥有权。

```
host1# metaset -s blue -r
```

```
host1# metaset -s blue
```

```
Set name = blue, Set number = 1
```

Host	Owner
host1	
host2	

Drive	Dbase
c1t6d0	Yes
c2t6d0	Yes

▼ 如何删除主机或磁盘集

如果要删除某个磁盘集，则该磁盘集不得包含任何磁盘，而且不得有任何其他主机连接到该磁盘集。删除最后一台主机会销毁该磁盘集。

- 1 要从磁盘集内删除主机或者删除磁盘集，请使用以下方法之一：
 - 从 Solaris Management Console 中增强的存储工具中，打开“磁盘集”节点。在要释放的磁盘集上单击鼠标右键，然后从弹出的菜单中选择“删除”。按照联机帮助中的说明操作。
 - 要删除主机，请使用以下形式的 `metaset` 命令：

```
metaset -s diskset-name -d -h hostname
```

-s *diskset-name* 指定执行 `metaset` 命令所针对磁盘集的名称。

-d 从磁盘集内删除主机。

-h *hostname* 指定要删除的主机的名称。

要删除磁盘集，请使用与上述 `metaset` 命令相同的格式。如果要删除某个磁盘集，则该磁盘集不得包含任何磁盘，而且任何其他主机都不得拥有该磁盘集。删除最后一台主机会销毁该磁盘集。

有关更多信息，请参见 `metaset(1M)` 手册页。

- 2 使用 `metaset` 命令来验证该主机是否已从磁盘集内删除。请注意，仅显示当前的（所属）主机。其他主机已被删除。

```
# metaset -s disk-set
```

示例 19-11 从磁盘集内删除主机

以下示例说明如何从磁盘集 `blue` 内删除主机 `host2`：

```
# metaset -s blue

Set name = blue, Set number = 1

Host                                Owner

  host1                             Yes

..host2

Drive                                Dbase

  c1t2d0                             Yes

  c1t3d0                             Yes

  c1t4d0                             Yes

  c1t5d0                             Yes

  c1t6d0                             Yes

  c2t1d0                             Yes

# metaset -s blue -d -h host2

# metaset -s blue
```

```
Set name = blue, Set number = 1
```

Host	Owner
host1	Yes

Drive	Dbase
c1t2d0	Yes
c1t3d0	Yes
c1t4d0	Yes
c1t5d0	Yes
c1t6d0	Yes
c2t1d0	Yes

示例 19-12 从磁盘集内删除最后一台主机

以下示例说明如何从磁盘集 blue 内删除最后一台主机：

```
host1# metaset -s blue -d -h host1

host1# metaset -s blue

metaset: host: setname "blue": no such set
```

导入磁盘集

导入磁盘集

使用 `metaimport` 命令，可以将磁盘集从一个系统导入到另一个系统。

▼ 如何列显有关可导入的磁盘集的报告

- 1 成为超级用户。
- 2 获取有关要导入的磁盘集的报告。

```
# metainport -r -v
```

-r 提供可导入到系统上的未配置磁盘集的报告。

-v 提供有关以下内容的详细信息：状态数据库 (metadb) 副本的位置；可导入到系统上的未配置磁盘集的磁盘状态。

示例 19-13 报告可导入的磁盘集

以下示例说明如何列显有关可导入的磁盘集的报告。metainport 命令的输出区分常规磁盘集和复制的磁盘集。

```
# metainport -r
```

```
# metainport -r
```

```
Drives in regular diskset including disk clt2d0:
```

```
clt2d0
```

```
clt3d0
```

```
More info:
```

```
metainport -r -v clt2d0
```

```
Import: metainport -s <newsetname> clt2d0
```

```
Drives in replicated diskset including disk clt4d0:
```

```
clt4d0
```

```
clt5d0
```

```
More info:
```

```
metainport -r -v clt4d0
```

```
Import: metainport -s <newsetname> clt4d0
```

```
# metainport -r -v clt2d0
```

```
Import: metaimport -s <newsetname> c1t2d0

Last update: Mon Dec 29 14:13:35 2003

Device      offset      length replica flags
c1t2d0      16          8192      a      u
c1t3d0      16          8192      a      u
```

▼ 如何将磁盘集从一个系统导入到另一个系统

1 成为超级用户。

2 验证磁盘集是否可以导入。

```
# metaimport -r -v
```

3 导入可用的磁盘集。

```
# metaimport -s diskset-name disk-name
```

-s *diskset-name* 指定所创建的磁盘集的名称。

disk-name 标识正在导入的磁盘集中包含状态数据库副本的磁盘 (c#t#d#)。

4 验证该磁盘集是否已导入。

```
# metaset -s diskset-name
```

示例 19-14 导入磁盘集

以下示例说明如何导入磁盘集。

```
# metaimport -s red c1t2d0
```

Drives in diskset including disk c1t2d0:

```
c1t2d0
```

```
c1t3d0
```

```
c1t8d0
```

More info:


```
metaimport -r -v clt2d0
```

```
# metaset -s red
```

```
Set name = red, Set number = 1
```

Host	Owner
------	-------

host1	Yes
-------	-----

Drive	Dbase
-------	-------

clt2d0	Yes
--------	-----

clt3d0	Yes
--------	-----

clt8d0	Yes
--------	-----

维护 Solaris Volume Manager（任务）

本章提供有关使用 Solaris Volume Manager 执行常规存储管理维护任务的信息。

以下是本章中信息的列表：

- 第 227 页中的 “Solaris Volume Manager 维护（任务列表）”
- 第 228 页中的 “查看 Solaris Volume Manager 配置”
- 第 235 页中的 “重命名卷”
- 第 238 页中的 “使用配置文件”
- 第 240 页中的 “更改 Solaris Volume Manager 缺省值”
- 第 240 页中的 “使用 `growfs` 命令扩展文件系统”
- 第 242 页中的 “在 RAID-1 和 RAID-5 卷中替换和启用组件的概述”

Solaris Volume Manager 维护（任务列表）

以下任务列表列出了维护 Solaris Volume Manager 所需的过程。

任务	说明	参考
查看 Solaris Volume Manager 配置	使用 Solaris Volume Manager GUI 或 <code>metastat</code> 命令查看系统配置。	第 228 页中的 “如何查看 Solaris Volume Manager 卷配置”
重命名卷	使用 Solaris Volume Manager GUI 或 <code>metarename</code> 命令重命名卷。	第 236 页中的 “如何重命名卷”
创建配置文件	使用 <code>metastat -p</code> 命令和 <code>metadb</code> 命令创建配置文件。	第 238 页中的 “如何创建配置文件”
根据配置文件初始化 Solaris Volume Manager	使用 <code>metainit</code> 命令根据配置文件初始化 Solaris Volume Manager。	第 238 页中的 “如何从配置文件中初始化 Solaris Volume Manager”

任务	说明	参考
扩展文件系统	使用 growfs 命令扩展文件系统。	第 241 页中的 “如何扩展文件系统”
启用组件	使用 Solaris Volume Manager GUI 或 metareplace 命令启用组件。	第 243 页中的 “启用组件”
替换组件	使用 Solaris Volume Manager GUI 或 metareplace 命令替换组件。	第 243 页中的 “使用其他可用组件来替换组件”

查看 Solaris Volume Manager 配置

提示 – metastat 命令不对输出进行排序。将 metastat -p 命令的输出通过管道传递给 sort 或 grep 命令，以获得更便于管理的配置列表。

▼ 如何查看 Solaris Volume Manager 卷配置

- 要查看卷配置，请使用以下方法之一：
 - 从 Solaris Management Console 中增强的存储工具中，打开“卷”节点。有关更多信息，请参见联机帮助。
 - 使用以下形式的 metastat 命令：

```
# metastat -p -i component-name
```

-p 指定以简洁摘要的形式显示输出。此输出适合在创建 md.tab 文件时使用。

-i 指定验证是否可以访问 RAID-1（镜像）卷、RAID-5 卷和热备件。

component-name 指定要查看的卷的名称。如果未指定卷名称，则将显示完整的组件列表。

示例 20-1 查看 Solaris Volume Manager 卷配置

以下示例显示 metastat 命令的输出。

```
# metastat

d50: RAID

      State: Okay
```

Interlace: 32 blocks

Size: 20985804 blocks

Original device:

Size: 20987680 blocks

Device	Start Block	Dbase	State	Reloc	Hot Spare
c1t4d0s5	330	No	Okay	Yes	
c1t5d0s5	330	No	Okay	Yes	
c2t4d0s5	330	No	Okay	Yes	
c2t5d0s5	330	No	Okay	Yes	
c1t1d0s5	330	No	Okay	Yes	
c2t1d0s5	330	No	Okay	Yes	

d1: Concat/Stripe

Size: 4197879 blocks

Stripe 0:

Device	Start Block	Dbase	Reloc
c1t2d0s3	0	No	Yes

d2: Concat/Stripe

Size: 4197879 blocks

Stripe 0:

Device	Start Block	Dbase	Reloc
c2t2d0s3	0	No	Yes

d80: Soft Partition

Device: d70

State: Okay

Size: 2097152 blocks

Extent	Start Block	Block count
0	1	2097152

d81: Soft Partition

Device: d70

State: Okay

Size: 2097152 blocks

Extent	Start Block	Block count
0	2097154	2097152

d70: Mirror

Submirror 0: d71

State: Okay

Submirror 1: d72

State: Okay

Pass: 1

Read option: roundrobin (default)

Write option: parallel (default)

Size: 12593637 blocks

d71: Submirror of d70

State: Okay

Size: 12593637 blocks

Stripe 0:

Device	Start	Block	Dbase	State	Reloc	Hot	Spare
clt3d0s3	0	No	Okay	Yes			

Stripe 1:

Device	Start	Block	Dbase	State	Reloc	Hot	Spare
clt3d0s4	0	No	Okay	Yes			

Stripe 2:

Device	Start	Block	Dbase	State	Reloc	Hot	Spare
clt3d0s5	0	No	Okay	Yes			

d72: Submirror of d70

State: Okay

Size: 12593637 blocks

Stripe 0:

Device	Start	Block	Dbase	State	Reloc	Hot	Spare
c2t3d0s3	0	No	Okay	Yes			

Stripe 1:

Device	Start Block	Dbase	State	Reloc	Hot Spare
c2t3d0s4	0	No	Okay	Yes	

Stripe 2:

Device	Start Block	Dbase	State	Reloc	Hot Spare
c2t3d0s5	0	No	Okay	Yes	

hsp010: is empty

hsp014: 2 hot spares

Device	Status	Length	Reloc
c1t2d0s1	Available	617652 blocks	Yes
c2t2d0s1	Available	617652 blocks	Yes

hsp050: 2 hot spares

Device	Status	Length	Reloc
c1t2d0s5	Available	4197879 blocks	Yes
c2t2d0s5	Available	4197879 blocks	Yes

hsp070: 2 hot spares

Device	Status	Length	Reloc
c1t2d0s4	Available	4197879 blocks	Yes
c2t2d0s4	Available	4197879 blocks	Yes

Device Relocation Information:

Device	Reloc	Device ID
c1t2d0	Yes	id1,sd@SSEAGATE_ST39204LCSUN9.0G3BV0N1S200002103AF29
c2t2d0	Yes	id1,sd@SSEAGATE_ST39204LCSUN9.0G3BV0P64Z00002105Q6J7
c1t1d0	Yes	id1,sd@SSEAGATE_ST39204LCSUN9.0G3BV0N1EM00002104NP2J
c2t1d0	Yes	id1,sd@SSEAGATE_ST39204LCSUN9.0G3BV0N93J000071040L3S
c0t0d0	Yes	id1,dad@s53554e575f4154415f5f53543339313430412525415933

示例 20-2 查看多 TB Solaris Volume Manager 卷

以下示例显示对多 TB 存储卷（11 TB）使用 `metastat` 命令的输出。

```
# metastat d0

d0: Concat/Stripe

Size: 25074708480 blocks (11 TB)

Stripe 0: (interlace: 32 blocks)

Device      Start Block  Dbase  Reloc
c27t8d3s0      0         No     Yes
c4t7d0s0     12288        No     Yes

Stripe 1: (interlace: 32 blocks)

Device      Start Block  Dbase  Reloc
c13t2d1s0     16384        No     Yes
c13t4d1s0     16384        No     Yes
c13t6d1s0     16384        No     Yes
c13t8d1s0     16384        No     Yes
c16t3d0s0     16384        No     Yes
```

c16t5d0s0	16384	No	Yes
c16t7d0s0	16384	No	Yes
c20t4d1s0	16384	No	Yes
c20t6d1s0	16384	No	Yes
c20t8d1s0	16384	No	Yes
c9t1d0s0	16384	No	Yes
c9t3d0s0	16384	No	Yes
c9t5d0s0	16384	No	Yes
c9t7d0s0	16384	No	Yes

Stripe 2: (interlace: 32 blocks)

Device	Start Block	Dbase	Reloc
c27t8d2s0	16384	No	Yes
c4t7d1s0	16384	No	Yes

Stripe 3: (interlace: 32 blocks)

Device	Start Block	Dbase	Reloc
c10t7d0s0	32768	No	Yes
c11t5d0s0	32768	No	Yes
c12t2d1s0	32768	No	Yes
c14t1d0s0	32768	No	Yes
c15t8d1s0	32768	No	Yes
c17t3d0s0	32768	No	Yes
c18t6d1s0	32768	No	Yes
c19t4d1s0	32768	No	Yes
c1t5d0s0	32768	No	Yes

c2t6d1s0	32768	No	Yes
c3t4d1s0	32768	No	Yes
c5t2d1s0	32768	No	Yes
c6t1d0s0	32768	No	Yes
c8t3d0s0	32768	No	Yes

其他参考信息

有关更多信息，请参见 `metastat(1M)` 手册页。

重命名卷

重命名卷的背景信息

使用 Solaris Volume Manager，可以遵从某些约束随时重命名大多数类型的卷。可以使用 Solaris Management Console 中增强的存储工具或命令行（`metarename(1M)` 命令）来重命名卷。

重命名卷或交换卷名的目的是为了便于管理卷名。例如，希望将所有文件系统挂载点安排为您希望的数字范围。您可能要重命名卷，以维护逻辑卷的命名方案，或允许事务卷使用与基础卷相同的名称。

注 – 事务卷在 Solaris Volume Manager 中不再有效。可以重命名事务卷，以将其替换。

重命名卷之前，请确保卷当前未被使用。对于文件系统，请确保文件系统未被挂载或用作 `swap`。使用原始设备的其他应用程序（如数据库）应该自备停止数据访问的方法。

重命名卷的特定注意事项包括以下几点：

- 可以重命名以下卷以外的任何卷：
 - 软分区
 - 直接生成软分区的卷
 - 当前用作日志设备的卷
 - 热备用池

- 可以重命名磁盘集中的卷。但是，不能通过重命名卷将卷从一个磁盘集移至另一个磁盘集。

交换卷名称

使用包含 `-x` 选项的 `metarename` 命令交换具有父子关系的卷的名称。有关更多信息，请参见第 236 页中的“如何重命名卷”和 `metarename(1M)` 手册页。现有卷的名称可以与其某个子组件的名称进行交换。例如，这种类型的交换可能出现在镜像与其某个子镜像之间。`metarename -x` 命令可以更加方便地镜像或取消镜像现有卷。

注 - 必须使用命令行来交换卷名称。此功能当前在 Solaris Volume Manager GUI 中不可用。但是，可以使用命令行或 GUI 来重命名卷。

需要重命名卷时，请考虑以下原则。

- 不能重命名当前正在使用的卷。此限制包括用作已挂载文件系统、`swap`、应用程序或数据库的活动存储区的卷。因此，使用 `metarename` 命令之前，请停止对将要重命名的卷进行的所有访问。例如，取消挂载已挂载的文件系统。
- 不能交换处于故障状态下的卷。
- 不能交换使用热备件替换的卷。
- 交换只能发生在具有直接父子关系的卷之间。
- 不能交换（或重命名）日志设备。解决方法是拆离日志设备，并连接具备所需名称的另一台日志设备。
- 只能交换卷。不能交换片或热备件。

▼ 如何重命名卷

开始之前 检查卷名称要求（第 42 页中的“卷名称”和第 235 页中的“重命名卷的背景信息”）。

1 取消挂载使用卷的文件系统。

```
# umount /filesystem
```

2 要重命名卷，请使用以下方法之一：

- 从 Solaris Management Console 中增强的存储工具中，打开“卷”节点。选择要重命名的卷。在对应的图标上单击鼠标右键。选择“属性”选项。然后，按照屏幕上的说明操作。有关更多信息，请参见联机帮助。
- 使用以下形式的 `metarename` 命令：

```
# metarename old-volume-name new-volume-name
```

old-volume-name 指定现有卷的名称。

new-volume-name 为现有卷指定新名称。

有关更多信息，请参见 `metarename(1M)` 手册页。

3 编辑 `/etc/vfstab` 文件，以引用新的卷名称（如有必要）。

4 重新挂载文件系统。

```
# mount /filesystem
```

示例 20-3 重命名用于文件系统的卷

在以下示例中，卷 `d10` 被重命名为 `d100`。

```
# umount /home
```

```
# metarename d10 d100
```

```
d10: has been renamed to d100
```

（编辑 `/etc/vfstab` 文件以使该文件系统引用新卷）

```
# mount /home
```

由于 `d10` 包含已挂载的文件系统，因此重命名卷之前，必须取消挂载该文件系统。如果该卷用于在 `/etc/vfstab` 文件中具有项的文件系统，则必须对该项进行更改，以引用新的卷名。

例如，如果 `/etc/vfstab` 文件包含文件系统的以下项：

```
/dev/md/dsk/d10 /dev/md/rdisk/d10 /docs home 2 yes -
```

请更改该项，使其显示为：

```
/dev/md/dsk/d100 /dev/md/rdisk/d100 /docs home 2 yes -
```

然后，重新挂载文件系统。

如果具有现有镜像或事务卷，则可以使用 `metarename -x` 命令来删除镜像或事务卷，并将数据保留在基础卷上。对于事务卷，只要主设备为卷（RAID-0、RAID-1 或 RAID-5 卷），就可以将数据保留在该卷上。

使用配置文件

Solaris Volume Manager 配置文件包含基本的 Solaris Volume Manager 信息，以及重新构造配置所必需的大多数数据。以下过程说明了如何使用这些文件。

▼ 如何创建配置文件

- ▶ 为 Solaris Volume Manager 环境定义所有相应参数后，请使用 `metastat -p` 命令创建 `/etc/lvm/md.tab` 文件。

```
# metastat -p > /etc/lvm/md.tab
```

此文件包含 `metainit` 命令和 `metabs` 命令使用的所有参数。如果需要设置多个类似环境或在出现系统故障后需要重新创建配置，请使用此文件。

有关 `md.tab` 文件的更多信息，请参见第 336 页中的“[md.tab 文件概述](#)”和 `md.tab(4)` 手册页。

▼ 如何从配置文件中初始化 Solaris Volume Manager



注意 - 请在以下情况下使用此过程：

- Solaris Volume Manager 配置全部丢失时
- 尚无任何配置且需要根据已保存的配置文件来创建配置时

有时，系统会丢失状态数据库中维护的信息。例如，如果在删除了所有的状态数据库副本后重新引导系统，就可能会丢失状态数据库中维护的信息。只要在丢失状态数据库后未创建卷，就可以使用 `md.cf` 或 `md.tab` 文件来恢复 Solaris Volume Manager 配置。

注 - `md.cf` 文件不维护有关活动热备件的信息。因此，如果在丢失 Solaris Volume Manager 配置时热备件正在使用中，则使用活动热备件的这些卷就可能被损坏。

有关这些文件的更多信息，请参见 `md.cf(4)` 和 `md.tab(4)` 手册页。

- 1 创建状态数据库副本。
有关更多信息，请参见第 70 页中的“[创建状态数据库副本](#)”。
- 2 创建或更新 `/etc/lvm/md.tab` 文件。

- 如果试图恢复已知的最新 Solaris Volume Manager 配置，请将 `md.cf` 文件复制到 `/etc/lvm/md.tab` 文件中。
- 如果您要根据已保留的 `md.tab` 文件副本创建新的 Solaris Volume Manager 配置，请将已保留的文件复制到 `/etc/lvm/md.tab` 文件中。

3 编辑“新”`/etc/lvm/md.tab` 文件并执行以下操作：

- 如果您要创建新的配置或要在系统崩溃后恢复配置，请将镜像配置为单向镜像。例如：

```
d80 -m d81 1
```

```
d81 1 1 clt6d0s3
```

如果镜像的子镜像大小不同，请确保使用此单向镜像的最小子镜像。否则，可能会丢失数据。

- 如果您要恢复现有配置且停止 Solaris Volume Manager 时没有报告任何问题，请将镜像配置保留为多向镜像。例如：

```
d70 -m d71 d72 1
```

```
d71 1 1 clt6d0s2
```

```
d72 1 1 clt5d0s0
```

- 使用 `-k` 选项指定 RAID-5 卷，以防止重新初始化设备。例如：

```
d45 -r clt3d0s5 clt3d0s3 clt3d0s4 -k -i 32b
```

有关更多信息，请参见 `metainit(1M)` 手册页。

4 使用以下一种形式的 `metainit` 命令检查 `/etc/lvm/md.tab` 文件项的语法，而不提交更改：

```
# metainit -n md.tab-entry
```

```
# metainit -n -a
```

与 `-n` 选项一起运行时，`metainit` 命令不会维护可能已创建的设备的虚拟状态，因此，如果所创建的卷依赖于其他不存在的卷，则使用 `-n` 选项运行该命令时将报告错误，即使该命令在不使用 `-n` 选项时能够成功执行也是如此。

`-n` 指定不实际创建设备。使用此选项验证结果是否和预期的一样。

`md.tab-entry` 指定要初始化的组件的名称。

`-a` 指定检查所有组件。

- 5 如果前面的步骤中未出现问题，请根据 `md.tab` 文件重新创建卷和热备用池。

```
# metainit -a
```

-a 指定激活 `/etc/lvm/md.tab` 文件中的项。

- 6 根据需要，使用 `metattach` 命令将单向镜像变成多向镜像。

```
# mettach mirror submirror
```

- 7 验证卷上的数据，以确认已正确重新构造配置。

```
# metastat
```

更改 Solaris Volume Manager 缺省值

在 Solaris 10 发行版中，Solaris Volume Manager 得到了增强，支持动态配置卷。无需再编辑 `/kernel/drv/md.conf` 文件中的 `nmd` 和 `md_nsets` 参数，便可根据需要动态创建新卷。

最大的 Solaris Volume Manager 配置值仍保持不变：

- 支持的最大卷数目为 8192。
- 支持的最大磁盘集数目为 32。

使用 growfs 命令扩展文件系统

扩展包含 UFS 文件系统的卷（意味着增加了更多空间）后，还需要扩展文件系统，以便识别扩展空间。必须使用 `growfs` 命令手动扩展文件系统。`growfs` 命令可扩展文件系统，即使已挂载的文件系统也可使用该命令进行扩展。但是，在 `growfs` 命令运行时不能对文件系统进行写入访问。

使用原始设备的应用程序（如数据库）必须自备用于合并扩展空间的方法。Solaris Volume Manager 不提供此功能。

`growfs` 命令可在扩展文件系统时对已挂载的文件系统进行“写锁定”。通过分阶段扩展文件系统，可以缩短对文件系统进行写锁定的时间长度。例如，要将 1 GB 文件系统扩展到 2 GB，则可以使用 `-s` 选项分阶段扩展文件系统的大小，每阶段增加 16 MB。此选项指定每个步级的新文件系统的总大小。

扩展期间，由于写锁定功能，不能对文件系统进行写入访问。系统将透明地暂停写入访问，并在 `growfs` 命令解除文件系统锁定时重新启动写入访问。读取访问不会受到影响。但是，在锁定生效期间不会保留访问时间。

扩展片和卷的背景信息

注 – 可以扩展 Solaris Volume Manager 卷。但是，不能缩小卷。

- 无论将卷用于文件系统、应用程序还是数据库，都可以扩展卷。可以扩展 RAID-0（条带化和串联）、RAID-1（镜像）卷以及 RAID-5 卷和软分区。
- 当现有文件系统处于使用状态时，可以串联包含该文件系统的卷。只要文件系统为 UFS 文件系统，就可以对该文件系统扩展（使用 `growfs` 命令），从而填充更大的空间。扩展文件系统时不会中断对数据的读取访问。
- 扩展文件系统后，由于 UFS 文件系统约束，不能缩小该文件系统。
- 使用原始设备的应用程序和数据库必须自动增加扩展空间的方法，以便应用程序可以识别它。Solaris Volume Manager 不提供此功能。
- 将组件添加到 RAID-5 卷时，组件将成为卷的串联体。新组件不包含奇偶校验信息。但是，新组件上的数据将受到对卷执行的整体奇偶校验计算的保护。
- 通过添加其他组件，可以扩展日志设备。无需运行 `growfs` 命令，因为在重新引导系统时 Solaris Volume Manager 可以自动识别增加的空间。
- 通过增加基础卷或片中的空间，可以扩展软分区。通过添加片，可以扩展所有其他卷。

▼ 如何扩展文件系统

开始之前 请查看第 45 页中的“创建 Solaris Volume Manager 组件的先决条件”。

1 查看与文件系统关联的磁盘空间。

```
# df -hk
```

有关更多信息，请参见 `df(1M)` 手册页。

2 扩展逻辑卷上的 UFS 文件系统。

```
# growfs -M /mount-point /dev/md/rdisk/volume-name
```

`-M/mount-point` 为要扩展的文件系统指定挂载点。

`/dev/md/rdisk/volume-name` 指定要扩展的卷名称。

有关更多信息，请参见以下示例和 `growfs(1M)` 手册页。

示例 20-4 扩展文件系统

在以下示例中，新的片已添加到卷 `d10` 中，该卷包含已挂载的文件系统 `/home2`。growfs 命令使用 `-M` 选项将挂载点指定为 `/home2`，该挂载点将被扩展到原始卷

/dev/md/rdsk/d10 中。growfs 命令完成后，文件系统将跨越整个卷。扩展文件系统前后，可以使用 df -hk 命令来验证磁盘总容量。

```
# df -hk

Filesystem            kbytes    used    avail capacity  Mounted on
...

/dev/md/dsk/d10        69047    65426         0   100%    /home2
...

# growfs -M /home2 /dev/md/rdsk/d10

/dev/md/rdsk/d10:      295200 sectors in 240 cylinders of 15 tracks, 82 sectors

      144.1MB in 15 cyl groups (16 c/g, 9.61MB/g, 4608 i/g)

super-block backups (for fsck -F ufs -o b=#) at:

    32, 19808, 39584, 59360, 79136, 98912, 118688, 138464, 158240, 178016, 197792,
    217568, 237344, 257120, 276896,

# df -hk

Filesystem            kbytes    used    avail capacity  Mounted on
...

/dev/md/dsk/d10        138703    65426    59407     53%    /home2
...
```

对于镜像卷，请始终对顶层卷运行 growfs 命令。请不要对子镜像或主设备运行该命令，即使已向子镜像或主设备添加空间也不要对其运行该命令。

在 RAID-1 和 RAID-5 卷中替换和启用组件的概述

Solaris Volume Manager 可以替换和启用 RAID-1（镜像）和 RAID-5 卷中的组件。

在 Solaris Volume Manager 术语中，替换组件是一种用系统中的可用组件替换子镜像或 RAID-5 卷中的选定组件的方式。可以将此过程视为逻辑替换，以区别于物理方式替换组件。有关更多信息，请参见第 243 页中的“使用其他可用组件来替换组件”。

启用组件就是“激活”或替换组件本身（也就是说，组件名称是相同的）。有关更多信息，请参见第 243 页中的“启用组件”。

注 – 因磁盘错误而进行恢复时，请浏览 `/var/adm/messages`，以了解出现了哪种错误。如果错误是暂时的，且磁盘本身没有问题，请尝试启用出故障的组件。也可以使用 `format` 命令来测试磁盘。

启用组件

存在以下任何一种情况时，都可以启用组件：

- Solaris Volume Manager 无法访问物理驱动器。例如，由于断电或驱动器电缆松动，可能会出现此问题。在这种情况下，Solaris Volume Manager 会将组件置于“Maintenance（维护）”状态。需要确保驱动器可以访问（恢复供电、重新连接电缆等），然后启用卷中的组件。
- 您怀疑物理驱动器具有的暂时问题与磁盘无关。仅通过启用处于“Maintenance（维护）”状态的组件，就可能修复该组件。如果通过启用组件未能解决问题，则需要执行以下操作之一：
 - 以物理方式替换磁盘驱动器并启用该组件
 - 使用系统中的另一个可用组件来替换该组件

以物理方式替换磁盘时，请一定要按照被替换磁盘的分区对新磁盘进行分区，从而确保每个已使用的组件上都有足够的空间。

注 – 请始终检查将要替换的磁盘上的状态数据库副本和热备件。替换磁盘之前，应删除处于错误状态的任何状态数据库副本。随后，在启用组件后，请重新创建同样大小的状态数据库副本。应采取同样的方式处理热备件。

使用其他可用组件来替换组件

使用可用但在系统中未使用的其他组件替换或交换现有组件时，可以使用 `metareplace` 命令。

存在以下任何一种情况时，都可以使用此命令：

- 磁盘驱动器有问题，但没有备用驱动器。不过，系统上的其他位置有可用组件。替换绝对有必要，但又不想关闭系统时，可能需要使用此策略。
- 在物理磁盘上看到软错误。

即使 Solaris Volume Manager 显示处于“Okay（正常）”状态下的镜像/子镜像或 RAID-5 卷，物理磁盘也可能报告软错误。使用其他可用组件替换有问题的组件后，即可执行预防性维护，并可防止出现硬错误。

- 需要执行性能调整。

评估组件的一种方式就是使用 Solaris Management Console 中增强的存储工具中可用的性能监视功能。例如，您可能发现，RAID-5 卷中的某个组件和平均负载比较高，但该组件仍处于“Okay（正常）”状态。要平衡卷上的负载，可以用磁盘上很少使用的组件来替换该组件。可以联机执行此类型的替换，而不会中断卷服务。

Maintenance（维护）和 Last Erred（最近出错）状态

当 RAID-1 或 RAID-5 卷中的组件遇到错误时，Solaris Volume Manager 会使组件处于“Maintenance（维护）”状态。不会对处于“Maintenance（维护）”状态的组件进一步执行任何读取或写入操作。

有时，组件会进入“Last Erred（最近出错）”状态。对于 RAID-1 卷，通常单向镜像会出现这种情况。卷将遇到错误。但是，没有可供读取的冗余组件。对于 RAID-5 卷，某个组件进入“Maintenance（维护）”状态且另一个组件出现故障后，会出现这种情况。第二个出现故障的组件将进入“Last Erred（最近出错）”状态。

当 RAID-1 卷或 RAID-5 卷中的组件处于“Last Erred（最近出错）”状态时，仍然会尝试对标记为“Last Erred（最近出错）”的组件执行 I/O。由于对 Solaris Volume Manager 而言，“Last Erred（最近出错）”组件包含最后一个完好的数据副本，因此会发生此 I/O 尝试。当组件处于“Last Erred（最近出错）”状态时，卷的行为与标准设备（磁盘）相同，卷将向应用程序返回 I/O 错误。通常，此时会丢失某些数据。

根据卷的类型，处理同一个卷中其他组件上的后续错误的方式会有所不同。

RAID-1 卷 RAID-1 卷可以允许许多组件处于“Maintenance（维护）”状态，且仍然可以读取和写入。如果组件处于“Maintenance（维护）”状态，则不会丢失任何数据。可以按照任意顺序安全地替换或启用组件。如果组件处于“Last Erred（最近出错）”状态，必须先替换处于“Maintenance（维护）”状态的组件，然后才能替换该组件。替换或启用处于“Last Erred（最近出错）”状态的组件通常意味着某些数据已丢失。修复镜像上的数据后，请确保对其进行验证。

RAID-5 卷 RAID-5 卷可允许单个组件处于“Maintenance（维护）”状态。可以安全地替换处于“Maintenance（维护）”的单个组件，而不会丢失数据。如果其他组件上出现错误，则该组件将进入“Last Erred（最近出错）”状态。此时，RAID-5 卷是只读设备。您需要执行某种类型的错误恢复，以使 RAID-5 卷的状态稳定，并降低数据丢失的可能性。如果 RAID-5 卷达到“Last Erred（最近出错）”状态，则很有可能已丢失数据。修复

RAID-5 卷上的数据后，请确保对其进行验证。

请始终先替换处于“Maintenance（维护）”状态的组件，然后再替换处于“Last Erred（最近出错）”状态的组件。替换和重新同步组件后，请使用 `metastat` 命令来验证其状态。然后，验证数据。

在 RAID-1 和 RAID-5 卷中替换和启用组件的背景信息

在 RAID-1 卷或 RAID-5 卷中替换组件时，请遵循以下原则：

- 请始终先替换处于“Maintenance（维护）”状态的组件，然后再替换处于“Last Erred（最近出错）”状态的组件。
- 替换和重新同步组件后，请使用 `metastat` 命令来验证卷的状态。然后，验证数据。替换或启用处于“Last Erred（最近出错）”状态的组件通常意味着某些数据已丢失。修复卷上的数据后，请确保对其进行验证。对于 UFS，请运行 `fsck` 命令以验证“元数据”（文件系统的结构）。然后，检查实际的用户数据。（实际上，用户必须检查其文件。）数据库或其他应用程序必须自备验证其内部数据结构的方式。
- 替换组件时，请始终检查状态数据库副本和热备件。替换物理磁盘之前，应删除处于错误状态的任何状态数据库副本。启用组件之前，应恢复状态数据库副本。同样的过程适用于热备件。
- 替换 RAID-5 卷的组件期间，可以采用以下两种方式来恢复数据：从当前正在使用的热备件恢复数据，或使用 RAID-5 奇偶校验（未使用热备件时）来恢复数据。
- 替换 RAID-1 卷的组件时，Solaris Volume Manager 将自动开始将新组件与该卷的其余组件重新同步。重新同步完成后，被替换的组件即可读取和写入。如果已使用热备件中的数据替换了出故障的组件，则该热备件将处于“Available（可用）”状态，且可用于其他热备件替换。
- 新的组件必须足够大，才能替换旧的组件。
- 为避免意外情况，替换处于“Last Erred（最近出错）”状态的设备之前，请备份所有数据。

Solaris Volume Manager 的最佳做法

本章通过使用 Solaris Volume Manager 的实际存储方案提供一般的最佳做法信息。在本章中，将首先介绍一个典型配置，然后进行分析，并提供满足相同需求所需的建议（“最佳做法”）配置。

本章包括以下信息：

- 第 247 页中的“部署小型服务器”
- 第 249 页中的“结合使用 Solaris Volume Manager 和网络存储设备”

部署小型服务器

分布式计算环境通常需要在多个位置部署类似或相同的服务器。这些环境包括 ISP，分布于不同地理位置的销售机构和电信服务提供商。分布式计算环境中的服务器可能提供以下某些服务：

- 路由器或防火墙服务
- 电子邮件服务
- DNS 高速缓存
- Usenet（网络新闻）服务器
- DHCP 服务
- 在各种不同位置提供的其他最佳服务

这些小型服务器具有几个共同的特征：

- 高可靠性要求
- 高可用性要求
- 常规硬件和性能要求

开始时，请考虑具有单个 SCSI 总线和两个内部磁盘的 Netra™ 服务器。此现有配置是分布式服务器的良好起点。可以方便地使用 Solaris Volume Manager 来镜像某些或所有片，进而提供冗余存储以帮助防止磁盘故障。有关此小型系统配置的示例，请参见下图。

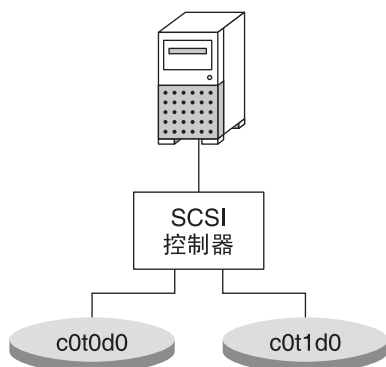


图 21-1 小型系统配置

此配置可能包括对根 (/)、/usr、swap、/var 和 /export 文件系统以及状态数据库副本（每个磁盘一个副本）的镜像。因而，任何镜像任一端的故障都不一定会导致系统故障。而且，最多可允许五个独立故障。但是，不能充分保护系统免遭磁盘或片故障的影响。各种潜在的故障都可能会导致整个系统出现故障，因而需要操作人员干预。

尽管此配置可帮助提供一些防止灾难性磁盘故障的保护措施，但还是暴露出主要的可能单点故障：

- 单个 SCSI 控制器是可能的故障点。如果控制器出现故障，系统将关闭，等待更换部件。
- 两个磁盘没有提供足够的状态数据库副本的分布。多数一致算法要求只有达到一半数量的状态数据库副本可用时，系统才能继续运行。此算法还要求只有当一半数量的副本再加上一个副本可用时，才能重新引导系统。因此，如果每个磁盘上各有一个状态数据库副本，且某个磁盘或包含副本的片出现故障，则无法重新引导系统。因此，镜像的根 (/) 文件系统就会无效。如果每个磁盘上有两个或更多个状态数据库副本，则一个片出现故障可能不是大问题。但是，磁盘故障仍然可能会阻止重新引导系统。如果各个磁盘上的副本数目不同，一个磁盘上的副本数目超过一半，而另一个磁盘的副本数目则不足一半。如果具有较少副本的磁盘出现了故障，则系统可以重新引导并继续工作。但是，如果具有较多副本的磁盘出现故障，则系统将立即崩溃。

“最佳做法”是再添加一个控制器以及一个硬盘驱动器来修改配置。生成的配置更具弹性。

结合使用 Solaris Volume Manager 和网络存储设备

Solaris Volume Manager 可以与网络存储设备完善配合，特别是那些提供可配置 RAID 级别和灵活选项的设备。通常，Solaris Volume Manager 与这类设备的组合产生的性能和灵活性优于单个产品。

一般情况下，请不要在提供冗余的任何硬件存储设备（如 RAID-1 和 RAID-5 卷）上建立 Solaris Volume Manager 的 RAID-5 卷。除非有非常特殊的情况，否则性能将受到影响。此外，冗余性或更高可用性方面的收益也会非常小。

另一方面，使用 RAID-5 卷配置基础硬件存储设备是非常有效的。这样做将为 Solaris Volume Manager 卷提供良好的基础。硬件 RAID-5 为 Solaris Volume Manager 的 RAID-1 卷、软分区或其他卷提供附加冗余性。

注 - 请勿配置类似的软件与硬件设备。例如，请不要在硬件 RAID-1 设备的顶部生成软件 RAID-1 卷。在硬件和软件中配置类似的设备会降低性能，但不会影响可靠性。

在基础硬件存储设备上生成的 Solaris Volume Manager 的 RAID-1 卷不是 RAID-1+0。Solaris Volume Manager 不能充分识别基础存储，因而无法提供 RAID-1+0 功能。

在 Solaris Volume Manager RAID-1 卷（在硬件 RAID-5 设备上生成的）的顶部配置软分区是非常灵活且有弹性的配置。

自上而下创建卷（概述）

本章提供有关 Solaris Volume Manager 自上而下创建卷的概念信息。

本章包含以下信息：

- 第 251 页中的“自上而下创建卷概述”
- 第 252 页中的“使用磁盘集实现自上而下创建卷”
- 第 252 页中的“自上而下创建卷的过程”
- 第 254 页中的“确定可用于自上而下创建卷的磁盘”

有关执行相关任务的信息，请参见第 23 章。

自上而下创建卷概述

通过自上而下创建卷，可以使用 `metassist` 命令自动创建 Solaris Volume Manager 卷配置。您不再需要手动完成对磁盘分区、创建 RAID-0 卷（作为子镜像）、创建热备用池和热备件，以及最终创建镜像的全过程。相反，您可以执行 `metassist` 命令来创建卷。Solaris Volume Manager 会为您执行其余的操作。

通过 `metassist` 命令，可以使用单个命令创建 Solaris Volume Manager 卷配置。可以根据**服务质量**指定卷的特征。服务质量特征是指不指定要在卷中使用的硬件组件的情况下，可以使用 `metassist` 命令的输入来提供以下信息：

- 卷大小
- 冗余级别，是指数据副本的数目
- 卷的数据路径的数目
- 故障恢复，指示卷是否与热备用池相关联

可以使用命令行选项或在命令行上指定的输入文件中按服务质量来指定卷。

在某些情况下，更具体地定义卷的特征或创建卷所应依据的约束是非常重要的。在这类情况下，还可以指定以下特征：

- 卷类型（例如，RAID-0（串联）卷或 RAID-0（条带化）卷）。

- 要在特定卷中使用的组件。
- 可用或不可用的组件。
- 要使用的组件数目。
- 特定于所创建的卷类型的详细信息。详细信息包括条带、镜像的读取策略以及类似的特征。

如果希望更详细地指定卷的名称、大小和组件，请使用输入文件。输入文件包括卷请求文件和卷规范文件。有关如何使用输入文件的更多信息，请参见第 252 页中的“[自上而下创建卷的过程](#)”。

最后，可以约束 `metassist` 命令使用（或不使用）特定磁盘或路径。

使用磁盘集实现自上而下创建卷

`metassist` 命令使用 Solaris Volume Manager 磁盘集来管理卷和用于自上而下创建卷的可用磁盘。对于任何给定的自上而下创建卷的过程，所有用作生成块的磁盘都必须位于磁盘集中，或可以添加到磁盘集。可以使用自上而下创建卷的过程在不同的磁盘集中创建卷。但是，可用的磁盘和组件受磁盘集功能的约束。

自上而下创建卷的过程

自上而下创建卷的过程通过提供以下过程来提供灵活性：

- 完全自动实现的端对端过程，通过该过程，可以指定所需的约束并且可以在命令完成后创建必要的卷
- 有关断点的更详尽的过程，可以在这些断点处向基于 XML 的文件进行写入

下图显示了 `metassist` 命令如何支持基于命令行输入和输入文件的端对端处理。图中还显示了 `metassist` 命令如何支持部分处理，通过此处理方式可提供基于文件的数据或检查卷的特征。

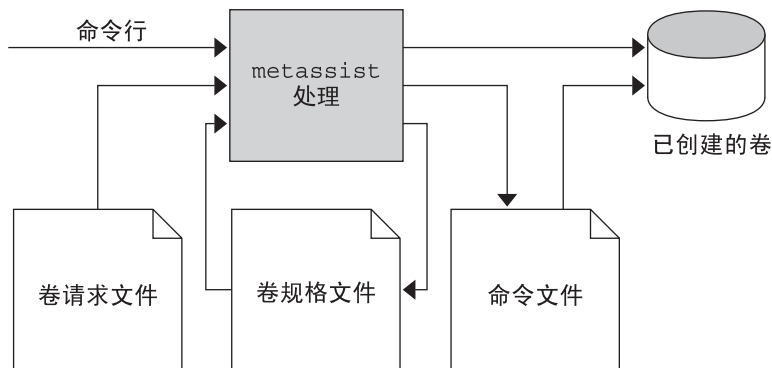


图 22-1 用于自上而下创建卷的处理选项

为实现无需介入的自动卷创建方法，请使用命令行指定所需的服务质量特征。`metassist` 命令可为您自动创建请求的卷。例如：

```
# metassist create -s storagepool -S 10Gb
```

此命令将在 `storagepool` 磁盘集中创建大小为 10 GB 的条带化卷。此命令将使用 `storagepool` 磁盘集中存在的可用存储空间。

或者，也可以使用 **卷请求文件** 来定义卷的特征。然后，可以使用 `metassist -F request-file` 命令来创建具有这些特征的卷。

可以使用 `metassist -d` 命令生成卷规范文件。如果需要，可以使用此文件来评估预定的实现并编辑文件。然后，可以将卷规范文件用作 `metassist` 命令的输入来创建卷。

最后，可以使用 `metassist -c` 命令来创建命令文件。**命令文件** 是用于实现 `metassist` 命令所指定的 Solaris Volume Manager 设备配置的 shell 脚本。可以使用此文件重复创建卷，并根据需要编辑文件。

使用 `metassist` 命令创建这些文件时，可了解 `metassist` 命令的作用及其做出决策的方式。此信息对于解答以下一些问题会非常有用：

- 为什么以特定的方式创建某个卷
- 为什么没有创建某个卷
- `metassist` 命令应创建哪些卷，但实际上却没有创建这些卷

确定可用于自上而下创建卷的磁盘

`metassist` 命令会检查磁盘，以确定哪些磁盘似乎未被使用。该命令将尝试保守地确定哪些磁盘可用。正在使用的任何磁盘或片都不能由 `metassist` 命令使用。`metassist` 命令将检查以下信息：

- 其他磁盘集中使用的磁盘
- 挂载的片
- 包含文件系统超级块的片，用于指示可挂载的文件系统
- 其他 Solaris Volume Manager 卷中使用的片

满足上述条件之一的任何片都不能用于自上而下创建卷。

自上而下创建卷（任务）

本章提供与使用 `metassist` 命令自上而下创建 Solaris Volume Manager 卷相关联的任务。

以下是本章中信息的列表：

- [第 255 页中的“自上而下创建卷（任务列表）”](#)
- [第 256 页中的“自上而下创建卷的先决条件”](#)
- [第 256 页中的“自动创建卷”](#)
- [第 261 页中的“使用 `metassist` 命令处理基于文件的数据”](#)
- [第 272 页中的“更改 `metassist` 命令的缺省行为”](#)

有关自上而下创建卷的概念性信息，请参见[第 22 章](#)。

自上而下创建卷（任务列表）

以下任务列表列出了使用 `metassist` 命令创建 Solaris Volume Manager 卷所需的过程。使用此命令可以基于服务质量特征指定卷，也可以创建多组分层卷。

任务	说明	参考
自动创建卷	允许您使用 <code>metassist</code> 命令创建一个或多个 Solaris Volume Manager 卷。 此外，允许您控制有关卷创建过程中的信息量，这些信息由 <code>metassist</code> 命令提供，以用于诊断或解决问题。	第 256 页中的“自动创建卷” 第 257 页中的“通过指定输出的详细程度来分析卷的创建过程”

任务	说明	参考
创建命令文件	帮助您使用 <code>metassist</code> 命令来创建 <code>shell</code> 脚本，以生成该命令所指定的卷。	第 269 页中的 “使用 <code>metassist</code> 命令创建卷配置文件”
用 <code>shell</code> 脚本创建卷	说明如何创建 <code>metassist</code> 命令使用其以前生成的 <code>shell</code> 脚本指定的 Solaris Volume Manager 卷。	第 268 页中的 “使用保存的 <code>Shell</code> 脚本（由 <code>metassist</code> 命令创建）创建卷”
创建卷配置文件	帮助您创建卷配置文件，并描述要创建的卷的特征。	第 269 页中的 “使用 <code>metassist</code> 命令创建卷配置文件”
更改卷缺省配置文件	允许您设置卷的缺省特征以自定义 <code>metassist</code> 命令的行为。	第 272 页中的 “更改卷缺省配置文件”

自上而下创建卷的先决条件

如果要使用 `metassist` 命令来自动创建卷和卷配置，则要求 Solaris Volume Manager 配置能够正常工作。在开始之前，应具备以下条件：

- 具有超级用户访问权限或者能够承担等效的基于角色的访问控制 (role-based access control, RBAC) 角色。有关更多信息，请参见《系统管理指南：基本管理》中的“成为超级用户 (root) 或承担角色”。
- 拥有专门针对您的系统分发的状态数据库副本。有关状态数据库副本的更多信息，请参见第 63 页中的 “关于 Solaris Volume Manager 状态数据库和副本”。
- 拥有可用于创建卷的磁盘。`metassist` 命令使用磁盘集来帮助管理存储。必须具有完全未使用的磁盘（或者现有的磁盘集），才能使用 `metassist` 命令创建新卷。有关磁盘可用性的更多信息，请参见第 254 页中的 “确定可用于自上而下创建卷的磁盘”。

除了上述最低要求，还要注意一点，那就是不要在 `/etc/inetd.conf` 文件中禁用 Solaris Volume Manager RPC 守护进程（`rpc.metad`、`rpc.metamhd` 和 `rpc.metamedd`）。在缺省情况下，这些守护进程配置为自动启动。它们必须保持启用状态，以便允许 Solaris Volume Manager 使用共享的磁盘集。

自动创建卷

使用 `metassist` 命令，可以基于服务质量标准创建 Solaris Volume Manager 卷或多组卷。仅使用 `metassist` 命令就可以创建卷，而不需要像以前在 Solaris Volume Manager 中那样使用一系列命令来创建卷。

可以使用 `metassist` 命令来直接创建 RAID-1（镜像）卷。因此，不必首先创建用作 RAID-1（镜像）卷组件的子镜像（串联或条带）。

通过指定输出的详细程度来分析卷的创建过程

在运行 `metassist` 命令时，可以指定输出的详细程度。详细程度高的输出有助于诊断问题，如确定磁盘为什么被（未被）选中以用于卷中，或者确定在尝试执行特定命令时为什么会失败。详细程度低的输出可减少必须检查的无关信息的量。

在指定输出的详细程度时，可以了解 `metassist` 命令的功能及其做出决定的方式。这些信息有助于解决下面的某些问题：

- 为什么以特定的方式创建某个卷
- 为什么没有创建某个卷
- `metassist` 命令应创建哪些卷，但实际上却没有创建这些卷

▼ 如何使用 `metassist` 命令来创建 RAID-1（镜像）卷

开始之前 请查看第 256 页中的“自上而下创建卷的先决条件”。

1 标识要在其上创建卷的可用存储器。

如果没有明确指定存储器，Solaris Volume Manager 会标识系统上未使用的存储器并使用它（如果适用的话）。如果您选择指定存储器—可以广泛地指定存储器（例如，控制器 1 上的所有存储器），也可以明确地指定存储器（例如，使用 `c1t4d2`，而不使用 `c1t4d1`），则 Solaris Volume Manager 会使用您指定的存储器。

2 使用 `metassist` 命令和适用于所执行任务的选项。

- 要使用命令行创建卷，请使用以下形式的 `metassist` 命令。

```
# metassist create -s diskset-name -f -r redundancy -a device1, device2... -S size -v verbosity
```

<code>create</code>	用于创建卷的子命令。
<code>-s diskset-name</code>	指定要用于卷的磁盘集的名称。
<code>-f</code>	指定该卷要与热备件相关联。
<code>-r redundancy</code>	指定要创建的冗余级别（数据副本的数量）。
<code>-a device1, device2...</code>	指定要用于创建卷的设备。
<code>-S size</code>	指定要创建的卷所采用的单位：KB（表示千字节）、MB（表示兆字节）、GB（表示千兆字节）或 TB（兆兆字节）。
<code>-v verbosity</code>	指定输出的详细程度。允许值介于 0（几乎无提示的输出）和 2（详细输出）之间。缺省详细程度是 1（中等详细程度的输出）。

- 如果要在创建卷时使用输入文件来指定卷的特征，请使用以下任一形式的 `metassist` 命令：

```
# metassist create [-v n] [-c] -F config_file
```

```
# metassist create [-v n] [-c | -d] -F request_file
```

- c 指定输出命令脚本，该命令脚本将实现指定的或所生成的卷配置。该命令脚本将不运行，而且处理将在该阶段停止。
- d 指定输出卷配置，该卷配置满足指定的或所生成的卷请求。将不生成或执行命令脚本，而且处理将在该阶段停止。
- F *config_file* | *request_file* 指定要处理的卷请求文件或卷配置文件。如果将 *config_file* 或 *request_file* 指定为连字符 (-)，系统将从标准输入中读取它。如果输入文件是卷配置文件，则不能指定 -d 选项。

卷配置文件描述要创建的卷的详细配置，而卷请求文件提供要生成的卷的特征。有关更多信息，请参见 [volume-config\(4\)](#)、[volume-config\(4\)](#) 和 [volume-request\(4\)](#) 手册页。
- v *verbosity* 指定输出的详细程度。允许值介于 0（几乎无提示的输出）和 2（详细输出）之间。缺省详细程度是 1（中等详细程度的输出）。

有关更多信息，请参见以下示例和 `metassist(1M)` 手册页。

3 在创建了卷之后，可以立即查看新卷。

```
# metastat -s diskset-name
```

示例 23-1 使用 metassist 命令创建双向镜像

以下示例说明如何创建大小为 10 MB 的双向镜像。metassist 命令标识未使用的磁盘并使用这些磁盘创建尽可能最佳的镜像。-s myset 参数指定将在 myset 磁盘集内创建卷。如有必要，请创建该磁盘集。

```
# metassist create -s myset -r 2 -S 10mb
```

示例 23-2 使用 metassist 命令创建双向镜像和热备件

以下示例说明如何使用 metassist 命令来创建大小为 10 MB 且具有热备件的双向镜像，热备件可提供额外的容错功能。-f 选项指定容错功能。

```
# metassist create -s myset -f -r 2 -S 10mb
```

示例 23-3 借助 metassist 命令使用特定的控制器创建条带

以下示例说明如何借助 metassist 命令使用控制器 1 上的磁盘创建条带。-a 选项指定可用的控制器。

```
# metassist create -s myset -a c1 -S 10mb
```

示例 23-4 使用 metassist 命令指定输出的详细程度

以下示例说明如何使用 metassist 命令来创建大小为 10 MB 且具有热备件的双向镜像，热备件可提供额外的容错功能。-f 选项指定容错功能。最后一个参数 (-v 2) 指定详细程度为 2（最大详细程度），它将尽可能多地提供有关 metassist 命令如何工作的信息。

```
# metassist create -s myset -f -r 2 -S 10mb -v 2
```

```
Scanning system physical device configuration...
```

```
These HBA/Controllers are known:.
```

```
c0                                /pci@1f,0/pci@1,1/ide@3
c1                                /pci@1f,0/pci@1/pci@2/SUNW,isp2@4
```

```
These disks are known:
```

```
c0t0d0                            id1,dad@AST34342A=_____VGD97101
c1t1d0                            id1,sd@SSEAGATE_ST39204LCSUN9.0G3BV0L88P000021097XNL
c1t2d0                            id1,sd@SSEAGATE_ST39102LCSUN9.0GLJW22867000019171JDF
c1t3d0                            id1,sd@SSEAGATE_ST39204LCSUN9.0G3BV0L7RV00007108TG0H
c1t4d0                            id1,sd@SSEAGATE_ST39204LCSUN9.0G3BV0LDFR000021087R1T
```

```

c1t5d0          id1,sd@SSEAGATE_ST39204LCSUN9.0G3BV0L0M200002109812L

c1t6d0          id1,sd@SSEAGATE_ST39204LCSUN9.0G3BV0L8K8000021087R0Z

.

.

.

(output truncated)

```

以下示例说明如何使用 `metassist` 命令来创建大小为 10 MB 且具有热备件的双向镜像，热备件可提供额外的容错功能。-f 选项指定容错功能。最后一个参数 (-v 0) 指定详细程度为 0（最小详细程度），在该命令运行时，它将提供几乎无提示的输出。

```

# metassist create -s myset -f -r 2 -S 10mb -v 0

myset/hsp000: Hotspare pool is setup

myset/hsp000: Hotspare is added

myset/d2: Concat/Stripe is setup

myset/d1: Concat/Stripe is setup

myset/d0: Mirror is setup

myset/d0: submirror myset/d1 is attached

```

示例 23-5 使用输入文件创建卷

以下示例说明如何借助 `metassist` 命令使用输入文件创建卷。

```
# metassist create -F request.xml
```

有关借助 `metassist` 命令使用输入文件创建卷的更多信息，请参见第 261 页中的“使用 `metassist` 命令处理基于文件的数据”。

使用 metassist 命令处理基于文件的数据

使用 metassist 命令，可以创建用来评估卷的特征或者实际创建卷的文件。

使用 metassist 命令创建命令文件（Shell 脚本）

通过运行带有 `-c` 参数的 metassist 命令，可以生成一个 Bourne shell 脚本，其中包含将用来创建卷配置的命令。使用此方法，可以在实际创建卷之前检查命令，甚至可以根据具体需要稍微修改脚本。

▼ 如何使用 metassist 命令来创建命令文件（Shell 脚本）

开始之前 请查看第 256 页中的“自上而下创建卷的先决条件”。

1 标识要在其上创建卷的可用存储器。

如果没有明确指定存储器，Solaris Volume Manager 会标识系统上未使用的存储器并使用它（如果适用的话）。如果您选择指定存储器—可以广泛地指定存储器（例如，控制器 1 上的所有存储器），也可以明确地指定存储器（例如，使用 `c1t4d2`，而不使用 `c1t4d1`），则 Solaris Volume Manager 会使用您指定的存储器。

2 使用 metassist 命令和适用于所执行任务的选项。

使用 `-c` 选项指定不应实际创建卷。

```
# metassist create -s diskset-name -f -r redundancy -a device1, device2... \
```

```
-S size -v verbosity [-c]
```

`create` 用于创建卷的子命令。

`-s diskset-name` 指定要用于卷的磁盘集的名称。

`-f` 指定该卷要与热备件相关联。

`-r redundancy` 指定要创建的冗余级别（数据副本的数量）。

`-a device1, device2...` 指定要用于创建卷的设备。

`-S size` 指定要创建的卷所采用的单位：KB（表示千字节）、MB（表示兆字节）、GB（表示千兆字节）或 TB（表示兆兆字节）。

`-v verbosity` 指定输出的详细程度。允许值介于 0（几乎无提示的输出）和 2（详细输出）之间。缺省详细程度是 1（中等详细程度的输出）。

-c 指定不应实际创建卷，而应将可用来创建指定配置的 shell 脚本发送到标准输出中。

注 -c 参数所需的 shell 脚本将发送到标准输出中，metassist 命令的其余输出将转至标准错误中。您可以在进行选择时重定向输出流。

有关更多信息，请参见以下示例和 metassist(1M) 手册页。

示例 23-6 使用 metassist 命令创建命令文件（Shell 脚本）

以下示例说明如何使用 metassist 命令来创建大小为 10 MB 且具有热备件的双向镜像，热备件可提供额外的容错功能。-f 选项指定容错功能。最后一个参数 (-c) 指定不应实际创建卷，而应将可用来创建指定配置的 shell 脚本发送到标准输出中。

```
# metassist create -s myset -f -r 2 -S 10mb -c
```

```
(output truncated)
```

```
.  
.  
.
```

```
Volume request completed successfully.
```

```
#!/bin/sh
```

```
#
```

```
# Environment
```

```
#
```

```
# Amend PATH
```

```
PATH="/usr/sbin:/usr/bin:$PATH"
```

```
export PATH
```

```
# Disk set name

diskset='myset'


#

# Functions

#


# Echo (verbose) and exec given command, exit on error
execho () {

    test -n "$verbose" && echo "$@"

    "$@" || exit

}


# Get full /dev/rdisk path of given slice
fullpath () {

    case "$1" in

        /dev/dsk/*/dev/did/dsk/*) echo "$1" | sed 's/dsk/rdsk/' ;;

        /*) echo "$1" ;;

        *) echo /dev/rdsk/"$1" ;;

    esac

}
```

```
# Run fmthard, ignore partboot error, error if output

fmthard_special () {

    ignore='Error writing partboot'

    out='fmthard "$@" 2>&1'

    result=$?

    echo "$out" |

    case "$out" in

        *"$ignore"*) grep -v "$ignore"; return 0 ;;

        '') return "$result" ;;

        *) cat; return 1 ;;

    esac >&2

}

#

# Main

#

# Verify root

if [ "$(id | sed 's/^[^()*(\[^\)]*\).*\/\1/' != root ]

then

    echo "This script must be run as root." >&2

    exit 1;

fi
```



```
# Check for verbose option

case "$1" in

    -v) verbose=1 ;;

    *) verbose= ;;

esac


# Does the disk set exist?

if metaset -s "$diskset" >/dev/null 2>&1

then

    # Take control of disk set

    execho metaset -s "$diskset" -t

else

    # Create the disk set

    autotakeargs=

    /usr/sbin/clinfo || autotakeargs='-A enable'

    execho metaset -s "$diskset" $autotakeargs -a -h 'uname -n | cut -f1 -d.'

fi


# Format slices

execho fmthard_special -d 7:0:0:0:0 'fullpath c1t3d0s7'

execho fmthard_special -d 7:0:0:0:0 'fullpath c1t6d0s7'

execho fmthard_special -d 7:0:0:0:0 'fullpath c1t4d0s7'
```

```
# Add disks to set

execho metaset -s "$diskset" -a clt3d0

execho metaset -s "$diskset" -a clt6d0

execho metaset -s "$diskset" -a clt4d0


# Format slices

execho fmthard_special -d 0:4:0:10773:17649765 'fullpath clt3d0s0'
execho fmthard_special -d 0:4:0:10773:17649765 'fullpath clt6d0s0'
execho fmthard_special -d 0:4:0:10773:17649765 'fullpath clt4d0s0'
execho fmthard_special -d 1:4:0:17660538:21546 'fullpath clt3d0s1'
execho fmthard_special -d 1:4:0:17660538:21546 'fullpath clt4d0s1'
execho fmthard_special -d 1:4:0:17660538:21546 'fullpath clt6d0s1'


# Does hsp000 exist?

metahs -s "$diskset" -i hsp000 >/dev/null 2>&1 || {

    # Create hsp hsp000

    execho metainit -s "$diskset" hsp000

}


# Add slices to hsp000

execho metahs -s "$diskset" -a hsp000 clt3d0s1


# Create concat d2

execho metainit -s "$diskset" d2 1 1 clt4d0s1
```

```
# Associate concat d2 with hot spare pool hsp000

execho metaparam -s "$diskset" -h hsp000 d2


# Create concat d1

execho metainit -s "$diskset" d1 1 1 c1t6d0s1


# Associate concat d1 with hot spare pool hsp000

execho metaparam -s "$diskset" -h hsp000 d1


# Create mirror d0

execho metainit -s "$diskset" d0 -m d2 1

execho metattach -s "$diskset" d0 d1

#
```

示例 23-7 使用 metassist 命令保存命令文件（Shell 脚本）

以下示例说明如何使用 metassist 命令来创建大小为 10 MB 且具有热备件的双向镜像，热备件可提供额外的容错功能。-f 选项指定容错功能。最后一个参数 (-c) 指定不应实际创建卷，而应将用来创建指定配置的 shell 脚本发送到标准输出中。该命令最后会重定向标准输出，以便创建 /tmp/metassist-shell-script.sh shell 脚本，该脚本以后可用来创建指定的卷。

```
# metassist create -s myset -f -r 2 -S 10mb -c > \

/tmp/metassist-shell-script.sh
```

使用保存的 **Shell** 脚本（由 metassist 命令创建）创建卷

使用 metassist 命令创建 shell 脚本之后，可以使用该脚本来创建在创建 shell 脚本时所指定的卷。



注意 – 在创建脚本时，由 metassist 命令创建的命令脚本在很大程度上依赖于在其上创建该脚本的系统的特定配置。在不同的系统上使用该脚本或者在更改系统配置之后使用该脚本可能会导致数据损坏或丢失。

▼ 如何执行保存的 metassist 命令 **Shell** 脚本

开始之前 请查看第 256 页中的“自上而下创建卷的先决条件”。

- 1 请确保自创建 shell 脚本以来系统配置未曾进行过更改，而且是在创建脚本的系统上执行脚本。

- 2 执行保存的 shell 脚本。

```
# sh ./metassist-shell-script-name
```

- 3 查看新卷。

```
# metastat -s diskset-name
```

示例 23-8 执行保存的 metassist 命令 Shell 脚本

以下示例说明如何借助 metassist 命令来使用 shell 脚本创建卷。

```
# sh ./tmp/metassist-shell-script.sh

myset/hsp000: Hotspare pool is setup

myset/hsp000: Hotspare is added

myset/d2: Concat/Stripe is setup

myset/d1: Concat/Stripe is setup

myset/d0: Mirror is setup

myset/d0: submirror myset/d1 is attached
```

使用 metassist 命令创建卷配置文件

通过运行带有 `-d` 参数的 `metassist` 命令，可以生成基于 XML 的卷配置文件，该配置文件详细指定要创建的卷及其组件，其中包括所有选项和有关这些卷的相关信息。检查该文件有助于了解 `metassist` 命令建议使用的配置。如果对该文件进行适当的更改，还可以微调配置。然后，将更改后的卷配置文件用作 `metassist` 命令的输入来实际创建卷。

▼ 如何使用 metassist 命令来创建卷配置文件

开始之前 请查看第 256 页中的“自上而下创建卷的先决条件”。

1 标识要在其上创建卷的可用存储器。

如果没有明确指定存储器，Solaris Volume Manager 会标识系统上未使用的存储器并使用它（如果适用的话）。如果您选择指定存储器—可以广泛地指定存储器（例如，控制器 1 上的所有存储器），也可以明确地指定存储器（例如，使用 `c1t4d2`，而不使用 `c1t4d1`），则 Solaris Volume Manager 会使用您指定的存储器。

2 使用 metassist 命令和适用于所执行任务的选项。

使用 `-d` 选项指定不应实际创建卷，而应将基于 XML 的卷配置文件发送到标准输出中：

```
# metassist create -s diskset-name -f -r redundancy -a device1, device2... \
```

```
-S size -v verbosity [-d]
```

`create` 用于创建卷的子命令。

`-s diskset-name` 指定要用于卷的磁盘集的名称。

`-f` 指定该卷要与热备件相关联。

`-r redundancy` 指定要创建的冗余级别（数据副本的数量）。

`-a device1, device2...` 指定要用于创建卷的设备。

`-S size` 指定要创建的卷所采用的单位：KB（表示千字节）、MB（表示兆字节）、GB（表示千兆字节）或 TB（兆兆字节）。

`-v verbosity` 指定输出的详细程度。允许值介于 0（几乎无提示的输出）和 2（详细输出）之间。缺省详细程度是 1（中等详细程度的输出）。

`-d` 指定不应实际创建卷。

注 - **-d** 参数所需的基于 XML 的卷配置文件将发送到标准输出中。但是，**metassist** 命令的其余输出将转至标准错误中。您可以在进行选择时重定向输出流。

有关更多信息，请参见以下示例和 **metassist(1M)** 手册页。

示例 23-9 使用 metassist 命令创建卷配置文件

以下示例说明如何使用 **metassist** 命令来创建大小为 10 MB 且具有热备件的双向镜像，热备件可提供额外的容错功能。**-f** 选项指定容错功能。最后一个参数 (**-d**) 指定不应实际创建卷，而应将最终可用来创建指定配置的卷配置文件发送到标准输出中。

```
# metassist create -s myset -f -r 2 -S 10mb -d
```

```
.(output truncated)
```

```
.  
.
```

```
Volume request completed successfully.
```

```
<?xml version="1.0"?>
```

```
<!DOCTYPE volume-config SYSTEM "/usr/share/lib/xml/dtd/volume-config.dtd">
```

```
<volume-config>
```

```
<diskset name="myset"/>
```

```
<disk name="c1t3d0"/>
```

```
<disk name="c1t6d0"/>
```

```
<disk name="c1t4d0"/>
```

```
<slice name="c1t3d0s7" sizeinblocks="0"/>
```

```
<slice name="c1t3d0s0" sizeinblocks="17649765" startsector="10773"/>
```

```

<slice name="c1t6d0s7" sizeinblocks="0"/>

<slice name="c1t6d0s0" sizeinblocks="17649765" startsector="10773"/>

<slice name="c1t4d0s7" sizeinblocks="0"/>

<slice name="c1t4d0s0" sizeinblocks="17649765" startsector="10773"/>

<hsp name="hsp000">

<slice name="c1t3d0s1" sizeinblocks="21546" startsector="17660538"/>

</hsp>

<mirror name="d0" read="ROUNDROBIN" write="PARALLEL" passnum="1">

<concat name="d2">

<slice name="c1t4d0s1" sizeinblocks="21546" startsector="17660538"/>

<hsp name="hsp000"/>

</concat>

<concat name="d1">

<slice name="c1t6d0s1" sizeinblocks="21546" startsector="17660538"/>

<hsp name="hsp000"/>

</concat>

</mirror>

</volume-config>

#

```

示例 23-10 使用 metassist 命令保存卷配置文件

以下示例说明如何使用 `metassist` 命令来创建大小为 10 MB 且具有热备件的双向镜像，热备件可提供额外的容错功能。`-f` 选项指定容错功能。最后一个参数 (`-d`) 指定不应实际创建卷，而应将最终可用来创建指定配置的卷配置文件发送到标准输出中。该命令最后会重定向标准输出，以便创建 `/tmp/metassist-volume-config.xml` 卷配置文件，该文件以后可用来创建指定的卷。

```
# metassist create -s myset -f -r 2 -S 10mb -d > \  
  
/tmp/metassist-volume-config.xml
```

更改 metassist 命令的缺省行为

可以使用卷缺省配置文件 (/etc/defaults/metassist.xml) 来更改 metassist 命令的缺省行为。通过更改缺省配置文件，可以明确地将特定磁盘或控制器排除或包括在考虑范围之内。还可以指定由 metassist 命令使用的大部分卷设置的要求。

/etc/defaults/metassist.xml 的格式由 /usr/share/lib/xml/dtd/volume-defaults.dtd 文档类型定义 (Document Type Definition, DTD) 来指定。该格式在 volume-defaults(4) 手册页中进行介绍。

更改卷缺省配置文件

编辑卷缺省配置文件 (/etc/defaults/metassist.xml) 以指定 metassist 命令的行为。

注 – 在编辑该文件时，必须确保该文件仍旧符合 /usr/share/lib/xml/dtd/volume-defaults.dtd 文档类型定义 (Document Type Definition, DTD)。如果该 XML 文件不符合 DTD，则 metassist 命令将失败，并且会出现一条错误消息。

示例 23-11 借助 metassist 命令使用更改后的缺省配置创建卷

在创建卷之前，请编辑 /etc/default/metassist.xml 文件以指定缺省设置，这些设置将应用于借助 metassist 命令创建的所有卷。在以下示例中，metassist 命令仅在控制器 c1 上创建卷，在创建条带时，仅创建恰好包含四个组件且交错值为 512KB 的条带。这些约束适用于 metassist 命令的所有应用场合，直到 /etc/default/metassist.xml 文件再次发生更改。

```
# cat /etc/default/metassist.xml  
  
<!DOCTYPE volume-defaults SYSTEM \  
  
"/usr/share/lib/xml/dtd/volume-defaults.dtd">  
  
<volume-defaults>
```


示例 23-11 借助 metassist 命令使用更改后的缺省配置创建卷 (续)

```
<available name="c1" />  
  
<stripe mincomp="4" maxcomp="4" interlace="512KB" ></stripe>  
  
</volume-defaults>
```

```
# metassist create -s myset -S 10Gb
```

metassist 命令会创建一个 10 GB 的条带，如 /etc/default/metassist.xml 文件中所指定，该条带将恰好使用四个片且交错值为 512 KB。

监视和错误报告（任务）

有时，Solaris Volume Manager 会遇到问题（如由于片级别的物理错误而无法写入卷）。出现问题后，Solaris Volume Manager 会更改卷的状态，以便系统管理员可以得到通知。但是，除非借助 Solaris Management Console 或者运行 `metastat` 命令在 Solaris Volume Manager GUI 中定期检查卷的状态，否则可能无法立即看到这些状态更改。

本章提供有关 Solaris Volume Manager 中提供的各种监视工具的信息。其中一个工具是 Solaris Volume Manager SNMP 代理，该代理是 Solstice Enterprise Agents™ 监视软件的子代理。除了可配置该工具使其报告 SNMP 陷阱以外，还可以创建一个 `shell` 脚本来不断监视多种 Solaris Volume Manager 功能。此 `shell` 脚本可以作为 `cron` 作业运行，而且可用于确定潜在的问题。

以下是本章中信息的列表：

- 第 275 页中的“Solaris Volume Manager 监视和报告（任务列表）”
- 第 276 页中的“配置 `mdmonitord` 命令以定期检查错误”
- 第 277 页中的“Solaris Volume Manager SNMP 代理概述”
- 第 278 页中的“配置 Solaris Volume Manager SNMP 代理”
- 第 281 页中的“Solaris Volume Manager SNMP 代理的限制”
- 第 281 页中的“用 `cron` 作业监视 Solaris Volume Manager”

Solaris Volume Manager 监视和报告（任务列表）

以下任务列表列出了管理 Solaris Volume Manager 错误报告所需的过程。

任务	说明	参考
配置 mdmonitord 守护进程，使其定期检查错误	通过编辑 /lib/svc/method/svc-mdmonitor 脚本来配置由 mdmonitord 守护进程使用的错误检查间隔。	第 276 页中的“配置 mdmonitord 命令以定期检查错误”
配置 Solaris Volume Manager SNMP 代理	编辑 /etc/snmp/conf 目录中的配置文件，以便 Solaris Volume Manager 正确地将陷阱发送给相应的系统。	第 278 页中的“配置 Solaris Volume Manager SNMP 代理”
用 cron 命令所运行的脚本监视 Solaris Volume Manager	创建新脚本或改编现有的脚本以检查错误，然后从 cron 命令运行该脚本。	第 281 页中的“用 cron 作业监视 Solaris Volume Manager”

配置 mdmonitord 命令以定期检查错误

Solaris Volume Manager 中包括 /usr/sbin/mdmonitord 守护进程。当磁盘出现故障时，Solaris Volume Manager 会检测到该故障并生成一个错误。此错误事件将触发 mdmonitord 守护进程，使其对 RAID-1（镜像）卷、RAID-5 卷和热备件执行检查。但是，您还可以将该程序配置为以指定的时间间隔不断检查错误。

▼ 如何配置 mdmonitord 命令以定期检查错误

编辑 /lib/svc/method/svc-mdmonitor 脚本，添加定期检查的时间间隔。

- 1 成为超级用户。
- 2 在所选编辑器中打开 /lib/svc/method/svc-mdmonitor 脚本。在该脚本中查找以下部分：

```
$MDMONITORD

error=$?

case $error in

0)      exit 0

        ;;

*)      echo "Could not start $MDMONITORD. Error $error."

        exit 0
```

- 3 更改以 `mdmonitord` 命令开头的行，具体操作为添加一个 `-t` 标志以及检查的时间间隔（以秒为单位）。

```
$MDMONITORD -t 3600

error=$?

case $error in

0)      exit 0

        ;;

*)      echo "Could not start $MDMONITORD. Error $error."

        exit 0

        ;;

esac
```

- 4 重新启动 `mdmonitord` 命令以激活所做的更改。

```
# svcadm restart system/mdmonitor
```

有关更多信息，请参见 `mdmonitord(1M)` 手册页。

Solaris Volume Manager SNMP 代理概述

Solaris Volume Manager SNMP 陷阱代理需要核心软件包 `SUNWlvmr`、`SUNWlvma` 的软件包以及 Solstice Enterprise Agents。核心软件包包括：

- `SUNWmibii`
- `SUNWsacom`
- `SUNWsadmi`
- `SUNWsasnm`

这些软件包是 Solaris 操作系统的一部分。缺省情况下会安装这些软件包，除非在安装时修改了软件包选项或者安装的是一组最少的软件包。要确认这些软件包是否可用，请使用 `pkginfo pkgname` 命令，正如在 `pkginfo SUNWsasnm` 中一样。确认了这五个软件包均可用之后，需要按照下一节中的说明配置 Solaris Volume Manager SNMP 代理。

配置 Solaris Volume Manager SNMP 代理

Solaris Volume Manager SNMP 代理在缺省情况下处于禁用状态。请使用以下过程启用 SNMP 陷阱。

无论何时升级 Solaris 操作系统，都可能需要编辑 `/etc/snmp/conf/enterprises.oid` 文件并再次附加步骤 6 中的行，然后重新启动 Solaris Enterprise Agents 服务器。

在完成此过程之后，系统将向指定的一台或多台主机发送 SNMP 陷阱。系统发送陷阱时，需要使用适当的 SNMP 监视器（如 Solstice Enterprise Agents 软件）来查看它们。

设置 `mdmonitord` 命令来定期探测系统，以帮助确保在出现问题时您能够收到陷阱。请参见第 276 页中的“配置 `mdmonitord` 命令以定期检查错误”。有关其他错误检查选项，另请参阅第 281 页中的“用 `cron` 作业监视 Solaris Volume Manager”。

▼ 如何配置 Solaris Volume Manager SNMP 代理

- 1 成为超级用户。
- 2 将 `/etc/snmp/conf/mdlogd.rsrc` 配置文件移到 `/etc/snmp/conf/mdlogd.rsrc` 中。

```
# mv /etc/snmp/conf/mdlogd.rsrc- /etc/snmp/conf/mdlogd.rsrc
```
- 3 编辑 `/etc/snmp/conf/mdlogd.acl` 文件以指定哪些主机应当接收 SNMP 陷阱。在该文件中查找以下内容：

```
trap = {
    {
        trap-community = SNMP-trap

        hosts = corsair

        {
            enterprise = "Solaris Volume Manager"

            trap-num = 1, 2, 3
        }
    }
}
```

更改包含 `hosts = corsair` 的行，以指定要接收 Solaris Volume Manager SNMP 陷阱的主机的名称。例如，如果要将 SNMP 陷阱发送到 `lexicon`，则应当将该行更改为 `hosts = lexicon`。如果要包括多台主机，请提供用逗号分隔的主机名列表，如 `hosts = lexicon, idiom`。

4 还需要编辑 /etc/snmp/conf/snmpdx.acl 文件以指定哪些主机应当接收 SNMP 陷阱。

查找以 `trap =` 开头的块并添加在上一步中添加的那些主机。该部分可能已用 `#` 注释掉。如果是这样，则必须去掉该部分中所需行开头的 `#`。陷阱部分中的其他行也已经注释掉。但是，您可以保留这些行，也可以为清楚起见而删除它们。在取消对所需行的注释并更新主机行之后，该部分看上去与以下内容类似：

```
#####

# trap parameters #

#####

trap = {

    {

        trap-community = SNMP-trap

        hosts =lexicon

        {

            enterprise = "sun"

            trap-num = 0, 1, 2-5, 6-16

        }

#     {

#         enterprise = "3Com"

#         trap-num = 4

#     }

#     {

#         enterprise = "snmp"

#         trap-num = 0, 2, 5

#     }

# }
```

```
# {  
  
#     trap-community = jerry-trap  
  
#     hosts = jerry, nanak, hubble  
  
#     {  
  
#         enterprise = "sun"  
  
#         trap-num = 1, 3  
  
#     }  
  
#     {  
  
#         enterprise = "snmp"  
  
#         trap-num = 1-3  
  
#     }  
  
    }  
  
}
```

注 – 请确保 /etc/snmp/conf/snmpdx.acl 文件中左括号和右括号的数量相同。

- 5 在 /etc/snmp/conf/snmpdx.acl 文件中，向已在上一步中取消注释的部分中添加一个新的 **Solaris Volume Manager** 部分。

```
    trap-community = SNMP-trap  
  
    hosts = lexicon  
  
    {  
  
        enterprise = "sun"  
  
        trap-num = 0, 1, 2-5, 6-16  
  
    }  
  
    {  
  
        enterprise = "Solaris Volume Manager"  
  
        trap-num = 1, 2, 3
```



```
}
```

请注意，所添加的四行将紧邻 enterprise = “sun” 块之后放置。

6 将下行附加到 /etc/snmp/conf/enterprises.oid 文件中：

```
"Solaris Volume Manager" "1.3.6.1.4.1.42.104"
```

7 停止并重新启动 Solstice Enterprise Agents 服务器。

```
# /etc/init.d/init.snmpdx stop
```

```
# /etc/init.d/init.snmpdx start
```

Solaris Volume Manager SNMP 代理的限制

对于管理员需要知道的所有 Solaris Volume Manager 问题，Solaris Volume Manager SNMP 代理并不能全部发送陷阱。具体来说，该代理仅在以下情况下才发送陷阱：

- RAID-1 和 RAID-5 子组件转入 “Needs Maintenance（需要维护）” 状态
- 交换热备件，将其投入使用
- 热备件开始重新同步
- 热备件完成了重新同步
- 镜像处于脱机状态
- 磁盘集由另一台主机提取，当前的主机崩溃

许多问题（如带有 RAID-0 卷或软分区的磁盘不可用）不会导致发送 SNMP 陷阱，即使在尝试读写该设备时也是如此。在这些情况下，通常会报告 SCSI 或 IDE 错误。但是，对于那些要报告给监视控制台的错误，必须由其他 SNMP 代理发送陷阱。

用 cron 作业监视 Solaris Volume Manager

▼ 如何自动检查卷中的错误

- ▶ 要自动检查 Solaris Volume Manager 配置是否有错误，请创建一个由 cron 实用程序定期运行的脚本。

以下示例显示一个可根据自己的需要改编和修改的脚本。

注 - 自动对 Solaris Volume Manager 进行错误检查的第一步是运行此脚本。您可能需要根据自己的配置修改此脚本。

```

#
#!/bin/ksh
#ident "@(#)metacheck.sh 1.3 96/06/21 SMI"
# ident='%Z%M% %I% %E% SMI'
#
# Copyright (c) 1999 by Sun Microsystems, Inc.
#
# metacheck
#
# Check on the status of the metadevice configuration. If there is a problem
# return a non zero exit code. Depending on options, send email notification.
#
# -h
# help
# -s setname
# Specify the set to check. By default, the 'local' set will be checked.
# -m recipient [recipient...]
# Send email notification to the specified recipients. This
# must be the last argument. The notification shows up as a short
# email message with a subject of
# "Solaris Volume Manager Problem: metacheck.who.nodename.setname"
# which summarizes the problem(s) and tells how to obtain detailed
# information. The "setname" is from the -s option, "who" is from
# the -w option, and "nodename" is reported by uname(1).
# Email notification is further affected by the following options:
# -f to suppress additional messages after a problem
# has been found.
# -d to control the suppression.
# -w to identify who generated the email.
# -t to force email even when there is no problem.
# -w who
# indicate who is running the command. By default, this is the
# user-name as reported by id(1M). This is used when sending
# email notification (-m).
# -f
# Enable filtering. Filtering applies to email notification (-m).
# Filtering requires root permission. When sending email notification
# the file /etc/lvm/metacheck.setname.pending is used to
# control the filter. The following matrix specifies the behavior
# of the filter:
#
# problem_found file_exists
# yes no Create file, send notification
# yes yes Resend notification if the current date
# (as specified by -d datefmt) is
# different than the file date.
# no yes Delete file, send notification
# that the problem is resolved.

```

```

#      no      no      Send notification if -t specified.
#
# -d datefmt
#   Specify the format of the date for filtering (-f). This option
#   controls the how often re-notification via email occurs. If the
#   current date according to the specified format (strftime(3C)) is
#   identical to the date contained in the
#   /etc/lvm/metacheck.setname.pending file then the message is
#   suppressed. The default date format is "%D", which will send one
#   re-notification per day.
# -t
#   Test mode. Enable email generation even when there is no problem.
#   Used for end-to-end verification of the mechanism and email addresses.
#
#
# These options are designed to allow integration of metacheck
# into crontab. For example, a root crontab entry of:
#
# 0,15,30,45 * * * * /usr/sbin/metacheck -f -w SVMcron \
# -d '\%D \%h' -m notice@example.com 2148357243.8333033@pager.example.com
#
# would check for problems every 15 minutes, and generate an email to
# notice@example.com (and send to an email pager service) every hour when
# there is a problem. Note the \ prior to the '%' characters for a
# crontab entry. Bounced email would come back to root@nodename.
# The subject line for email generated by the above line would be
# Solaris Volume Manager Problem: metacheck.SVMcron.nodename.local
#

# display a debug line to controlling terminal (works in pipes)
decho()
{
    if [ "$debug" = "yes" ] ; then
        echo "DEBUG: $"      < /dev/null > /dev/tty 2>&1
    fi
}

# if string $1 is in $2-* then return $1, else return ""
strstr()
{
    typeset    look="$1"
    typeset    ret=""

    shift

    # decho "strstr LOOK .$look. FIRST .$1."
    while [ $# -ne 0 ] ; do
        if [ "$look" = "$1" ] ; then
            ret="$look"
        fi
    done
}

```

```
        fi
        shift
    done
    echo "$ret"
}

# if string $1 is in $2-* then delete it. return result
strdstr()
{
    typeset    look="$1"
    typeset    ret=""

    shift
#   decho "strdstr LOOK .$look. FIRST .$1."
    while [ $# -ne 0 ] ; do
        if [ "$look" != "$1" ] ; then
            ret="$ret $1"
        fi
        shift
    done
    echo "$ret"
}

merge_continued_lines()
{
    awk -e '\
BEGIN { line = ""; } \
$NF == "\\\" { \
    $NF = ""; \
    line = line $0; \
    next; \
} \
$NF != "\\\" { \
    if ( line != "" ) { \
        print line $0; \
        line = ""; \
    } else { \
        print $0; \
    } \
}'
}

# trim out stuff not associated with metadevices
find_meta_devices()
{
    typeset    devices=""

#   decho "find_meta_devices .$*."
}
```

```

while [ $# -ne 0 ] ; do
case $1 in
d+([0-9]) )    # metadvice name
    devices="$devices $1"
    ;;
esac
shift
done
echo "$devices"
}

# return the list of top level metadvice
toplevel()
{
    typeset    comp_meta_devices=""
    typeset    top_meta_devices=""
    typeset    devices=""
    typeset    device=""
    typeset    comp=""

    metastat$setarg -p | merge_continued_lines | while read line ; do
    echo "$line"
    devices='find_meta_devices $line'
    set -- $devices
    if [ $# -ne 0 ] ; then
        device=$1
        shift
        # check to see if device already referred to as component
        comp='strstr $device $comp_meta_devices'
        if [ -z $comp ] ; then
            top_meta_devices="$top_meta_devices $device"
        fi
        # add components to component list, remove from top list
        while [ $# -ne 0 ] ; do
            comp=$1
            comp_meta_devices="$comp_meta_devices $comp"
            top_meta_devices='strdstr $comp $top_meta_devices'
            shift
        done
        fi
    done > /dev/null 2>&1
    echo $top_meta_devices
}

#
# - MAIN
#
METAPATH=/usr/sbin

```

```

PATH=/usr/bin:$METAPATH
USAGE="usage: metacheck [-s setname] [-h] [[-t] [-f [-d datefmt]] \
    [-w who] -m recipient [recipient...]]"

datefmt="%D"
debug="no"
filter="no"
mflag="no"
set="local"
setarg=""
testarg="no"
who='id | sed -e 's/^uid=[0-9][0-9]*(//\' -e 's/).*/\''

while getopts d:Dfms:tw: flag
do
    case $flag in
        d)    datefmt=$OPTARG;
            ;;
        D)    debug="yes"
            ;;
        f)    filter="yes"
            ;;
        m)    mflag="yes"
            ;;
        s)    set=$OPTARG;
            if [ "$set" != "local" ] ; then
                setarg=" -s $set";
            fi
            ;;
        t)    testarg="yes";
            ;;
        w)    who=$OPTARG;
            ;;
        \?)    echo $USAGE
            exit 1
            ;;
        esac
done

# if mflag specified then everything else part of recipient
shift `expr $OPTIND - 1`
if [ $mflag = "no" ] ; then
    if [ $# -ne 0 ] ; then
        echo $USAGE
        exit 1
    fi
else
    if [ $# -eq 0 ] ; then

```

```

        echo $USAGE
        exit 1
    fi
fi
recipients="$*"

curdate_filter='date +%datefmt'
curdate='date'
node='uname -n'

# establish files
msg_f=/tmp/metacheck.msg.$$
msgs_f=/tmp/metacheck.msgs.$$
metastat_f=/tmp/metacheck.metastat.$$
metadb_f=/tmp/metacheck.metadb.$$
metahs_f=/tmp/metacheck.metahs.$$
pending_f=/etc/lvm/metacheck.$set.pending
files="$metastat_f $metadb_f $metahs_f $msg_f $msgs_f"

rm -f $files                                > /dev/null 2>&1
trap "rm -f $files > /dev/null 2>&1; exit 1" 1 2 3 15

# Check to see if metadb is capable of running
have_metadb="yes"
metadb$setarg                                > $metadb_f 2>&1
if [ $? -ne 0 ] ; then
    have_metadb="no"
fi
grep "there are no existing databases"        < $metadb_f    > /dev/null 2>&1
if [ $? -eq 0 ] ; then
    have_metadb="no"
fi
grep "/dev/md/admin"                          < $metadb_f    > /dev/null 2>&1
if [ $? -eq 0 ] ; then
    have_metadb="no"
fi

# check for problems accessing metadbs
retval=0
if [ "$have_metadb" = "no" ] ; then
    retval=1
    echo "metacheck: metadb problem, can't run '$METAPATH/metadb$setarg'" \
        >> $msgs_f
else
    # snapshot the state
    metadb$setarg 2>&1 | sed -e '1d' | merge_continued_lines    > $metadb_f
    metastat$setarg 2>&1 | merge_continued_lines                > $metastat_f
    metahs$setarg -i 2>&1 | merge_continued_lines              > $metahs_f

```

```

#
# Check replicas for problems, capital letters in the flags
# indicate an error, fields are separated by tabs.
#
problem='awk < $metadb_f -F\t '{if ($1 ~ /[A-Z]/) print $1;}}'
if [ -n "$problem" ] ; then
    retval='expr $retval + 64'
    echo "\
metacheck: metadb problem, for more detail run:\n\t$METAPATH/metadb$setarg -i" \
        >> $msgs_f
fi

#
# Check the metadevice state
#
problem='awk < $metastat_f -e \
    '/State:/ {if ($2 != "Okay" && $2 != "Resyncing") print $0;}}'
if [ -n "$problem" ] ; then
    retval='expr $retval + 128'
    echo "\
metacheck: metadevice problem, for more detail run:" \
        >> $msgs_f

# refine the message to toplevel metadevices that have a problem
top='toplevel'
set -- $top
while [ $# -ne 0 ] ; do
    device=$1
    problem='metastat $device | awk -e \
        '/State:/ {if ($2 != "Okay" && $2 != "Resyncing") print $0;}}'
    if [ -n "$problem" ] ; then
        echo "\t$METAPATH/metastat$setarg $device" >> $msgs_f
        # find out what is mounted on the device
        mp='mount|awk -e '/\dev\md\dsk\/'$device'[ \t]/{print $1;}}'
        if [ -n "$mp" ] ; then
            echo "\t\t$mp mounted on $device" >> $msgs_f
        fi
    fi
    shift
done
fi

#
# Check the hotspares to see if any have been used.
#
problem=""
grep "no hotspare pools found" < $metahs_f > /dev/null 2>&1

```



```

        if [ $? -ne 0 ] ; then
        problem='awk < $metahs_f -e \
        '/blocks/ { if ( $2 != "Available" ) print $0;}'
        fi
        if [ -n "$problem" ] ; then
        retval='expr $retval + 256'
        echo "\
metacheck: hot spare in use, for more detail run:\n\t$METAPATH/metahs$setarg -i" \
        >> $msgs_f
        fi
    fi

# If any errors occurred, then mail the report
if [ $retval -ne 0 ] ; then
    if [ -n "$recipients" ] ; then
        re=""
        if [ -f $pending_f ] && [ "$filter" = "yes" ] ; then
            re="Re: "
            # we have a pending notification, check date to see if we resend
            penddate_filter='cat $pending_f | head -1'
            if [ "$curdate_filter" != "$penddate_filter" ] ; then
                rm -f $pending_f > /dev/null 2>&1
            else
                if [ "$debug" = "yes" ] ; then
                    echo "metacheck: email problem notification still pending"
                    cat $pending_f
                fi
            fi
        fi
        if [ ! -f $pending_f ] ; then
            if [ "$filter" = "yes" ] ; then
                echo "$curdate_filter\n\tDate:$curdate\n\tTo:$recipients" \
                > $pending_f
            fi
            echo "\
Solaris Volume Manager: $node: metacheck$setarg: Report: $curdate" >> $msg_f
            echo "\
-----" >> $msg_f
            cat $msg_f $msgs_f | mailx -s \
            "${re}Solaris Volume Manager Problem: metacheck.$who.$set.$node" $recipients
        fi
        else
        cat $msgs_f
        fi
    else
        # no problems detected,
        if [ -n "$recipients" ] ; then
            # default is to not send any mail, or print anything.

```

```
    echo "\
Solaris Volume Manager: $node: metacheck$setarg: Report: $curdate"      >> $msg_f
    echo "\
-----" >> $msg_f
    if [ -f $pending_f ] && [ "$filter" = "yes" ] ; then
        # pending filter exists, remove it and send OK
        rm -f $pending_f > /dev/null 2>&1
        echo "Problem resolved" >> $msg_f
        cat $msg_f | mailx -s \
            "Re: Solaris Volume Manager Problem: metacheck.$who.$node.$set" $recipients
    elif [ "$testarg" = "yes" ] ; then
        # for testing, send mail every time even though there is no problem
        echo "Messaging test, no problems detected" >> $msg_f
        cat $msg_f | mailx -s \
            "Solaris Volume Manager Problem: metacheck.$who.$node.$set" $recipients
    fi
else
    echo "metacheck: Okay"
fi

rm -f $files > /dev/null 2>&1
exit $retval
```

有关使用 cron 实用程序来调用脚本的信息，请参见 cron(1M) 手册页。

对 Solaris Volume Manager 进行疑难解答 (任务)

本章介绍如何解决与 Solaris Volume Manager 相关的问题。除提供一般的疑难解答原则以外，本章还提供用来解决某些已知问题的特定过程。

本章包括以下信息：

- 第 291 页中的 “对 Solaris Volume Manager 进行疑难解答（任务列表）”
- 第 292 页中的 “系统疑难解答概述”
- 第 293 页中的 “替换磁盘”
- 第 296 页中的 “从磁盘移动问题中恢复”
- 第 297 页中的 “升级到 Solaris 10 发行版之后设备 ID 的改变”
- 第 299 页中的 “从引导问题中恢复”
- 第 310 页中的 “从状态数据库副本故障中恢复”
- 第 314 页中的 “从软分区问题中恢复”
- 第 317 页中的 “从另一个系统中恢复存储”
- 第 329 页中的 “从磁盘集问题中恢复”
- 第 331 页中的 “使用 `ufsdump` 命令对已挂载的文件系统执行备份”
- 第 333 页中的 “执行系统恢复”

本章介绍 Solaris Volume Manager 的一些问题及其相应的解决方案。本章无意面面俱到，仅提供一些常用的方案和恢复过程。

对 Solaris Volume Manager 进行疑难解答（任务列表）

以下任务列表列出了对 Solaris Volume Manager 进行疑难解答所必需的一些过程。

任务	说明	参考
替换出现故障的磁盘	替换磁盘，然后更新新磁盘上的状态数据库副本和逻辑卷。	第 293 页中的 “如何替换出现故障的磁盘”

任务	说明	参考
从磁盘移动问题中恢复	将磁盘恢复到原始位置或者与产品支持人员联系。	第 296 页中的“从磁盘移动问题中恢复”
从错误的 <code>/etc/vfstab</code> 项中恢复	针对镜像执行 <code>fsck</code> 命令，然后编辑 <code>/etc/vfstab</code> 文件，以便系统能够正确引导。	第 300 页中的“如何从错误的 <code>/etc/vfstab</code> 项中恢复”
从引导设备故障中恢复	从另一个子镜像引导。	第 303 页中的“如何从引导设备故障中恢复”
从状态数据库副本不足中恢复	使用 <code>metadb</code> 命令删除不可用的副本。	第 310 页中的“如何从状态数据库副本不足中恢复”
恢复已丢失软分区的配置数据	使用 <code>metarecover</code> 命令恢复软分区的配置数据。	第 314 页中的“如何恢复软分区的配置数据”
从修复的磁盘中恢复 Solaris Volume Manager 配置	将磁盘连接到新系统，并让 Solaris Volume Manager 从现有的状态数据库副本重新生成配置。	第 317 页中的“如何从本地磁盘集恢复存储”
从另一个系统中恢复存储	将存储从已知磁盘集导入另一个系统。	第 317 页中的“从另一个系统中恢复存储”
清除不可访问的磁盘集。	使用 <code>metaset</code> 命令清除不能获取或使用的磁盘集的内容。	第 329 页中的“从磁盘集问题中恢复”
恢复存储在 Solaris Volume Manager 卷上的系统配置。	使用 Solaris OS 安装介质恢复存储在 Solaris Volume Manager 卷上的系统配置。	第 333 页中的“执行系统恢复”

系统疑难解答概述

对系统进行疑难解答的先决条件

要解决与 Solaris Volume Manager 相关的存储管理问题，需要具备以下条件：

- 具有超级用户权限
- 拥有所有数据的最新备份

对 Solaris Volume Manager 进行疑难解答的一般原则

在解决 Solaris Volume Manager 问题时，应当准备好以下信息：

- `metadb` 命令的输出

- `metastat` 命令的输出
- `metastat -p` 命令的输出
- `/etc/vfstab` 文件的备份副本
- `/etc/lvm/mddb.cf` 文件的备份副本
- `prtvtoc` 命令（SPARC® 系统）或 `fdisk` 命令（基于 x86 的系统）的磁盘分区信息
- 系统上 Solaris 的版本
- 已安装的 Solaris 修补程序的列表
- 已安装的 Solaris Volume Manager 修补程序的列表

提示 – 无论何时更新 Solaris Volume Manager 配置或者对系统进行其他与存储或操作系统相关的更改，都会生成此配置信息的新副本。还可以使用 `cron` 作业自动生成此信息。

一般疑难解答方法

尽管无法通过一个过程来评估所有的 Solaris Volume Manager 问题，但是以下过程提供的一般方法可能会有所帮助。

1. 收集有关当前配置的信息。
2. 查看当前状态的指示器（包括 `metastat` 和 `metadb` 命令的输出）。此信息应指示哪个组件存在故障。
3. 检查硬件是否存在明显的故障点：
 - 所有电缆的连接是否正确？
 - 最近是否停过电？
 - 是否曾经更改或添加过设备？

替换磁盘

本节介绍如何在 Solaris Volume Manager 环境中替换磁盘。



注意 – 如果出现故障的磁盘上有软分区，或者在出现故障的磁盘上构建的卷上有软分区，则必须将新磁盘放在相同的物理位置中。而且，还要将相同的 `cntndn` 编号用作要替换的磁盘。

▼ 如何替换出现故障的磁盘

- 1 通过检查 `/var/adm/messages` 文件以及 `metastat` 命令的输出，确定要替换的出现故障的磁盘。

2 查找可能放在出现故障的磁盘上的所有状态数据库副本。

可使用 `metadb` 命令查找这些副本。

`metadb` 命令会报告出现故障的磁盘上状态数据库副本的错误。在此示例中，`c0t1d0` 是有问题的设备。

```
# metadb

      flags      first blk      block count

a m    u         16           1034      /dev/dsk/c0t0d0s4

a      u        1050           1034      /dev/dsk/c0t0d0s4

a      u        2084           1034      /dev/dsk/c0t0d0s4

W  pc luo        16           1034      /dev/dsk/c0t1d0s4

W  pc luo       1050           1034      /dev/dsk/c0t1d0s4

W  pc luo       2084           1034      /dev/dsk/c0t1d0s4
```

输出显示，在每个本地磁盘（`c0t0d0` 和 `c0t1d0`）的片 4 上都有三个状态数据库副本。`c0t1d0s4` 片的标志字段中的 `W` 指示设备有写入错误。`c0t0d0s4` 片上的三个副本仍处于良好状态。

3 记录状态数据库副本所在片的名称以及状态数据库副本的数量。然后删除状态数据库副本。

状态数据库副本的数量可通过计算某个片在 `metadb` 命令输出中出现的次数来获得。在此示例中，删除了 `c0t1d0s4` 上存在的三个状态数据库副本。

```
# metadb -d c0t1d0s4
```



注意 - 在删除有问题的状态数据库副本之后，如果只剩下三个或更少的副本，请先[添加更多的状态数据库副本](#)，然后再继续执行操作。这样有助于确保配置信息保持不变。

4 查找和删除出现故障的磁盘上的所有热备件。

可使用 `metastat` 命令来查找热备件。在此示例中，热备用池 `hsp000` 包括 `c0t1d0s6`，该片随后将从热备用池中删除。

```
# metahs -d hsp000 c0t1d0s6
```

```
hsp000: Hotspare is deleted
```

5 替换出现故障的磁盘。

此步骤可能需要使用 `cfgadm` 命令、`luxadm` 命令或其他适用于您的硬件和环境的命令。在执行此步骤时，请确保按照硬件文档中的过程正确地处理该磁盘的 Solaris 状态。

6 对新磁盘重新分区。

使用 `format` 命令或 `fmthard` 命令，用与出现故障的磁盘相同的片信息对新磁盘进行分区。如果拥有出现故障的磁盘的 `prtvtoc` 输出，则可以用 `fmthard -s /tmp/failed-disk-prtvtoc-output` 命令格式化替换磁盘。

7 如果删除了状态数据库副本，请向相应的片中重新添加相同的编号。

此示例中使用的是 `/dev/dsk/c0t1d0s4`。

```
# metadb -a -c 3 c0t1d0s4
```

8 如果磁盘上的所有片都是 RAID-5 卷或 RAID-0 卷的组件，而这些卷又是 RAID-1 卷的子镜像，请针对每个片运行 `metareplace -e` 命令。

此示例中使用的是 `/dev/dsk/c0t1d0s4` 和镜像 `d10`。

```
# metareplace -e d10 c0t1d0s4
```

9 如果直接在替换磁盘的片上构建软分区，请针对每个包含软分区的片运行 `metarecover -m -p` 命令。此命令会在磁盘上重新生成分区扩展表头。

在此示例中，`/dev/dsk/c0t1d0s4` 需要在磁盘上重新生成软分区标记。系统会扫描该片并基于状态数据库副本中的信息重新应用这些标记。

```
# metarecover c0t1d0s4 -m -p
```

10 如果磁盘上的所有软分区都是 RAID-5 卷或 RAID-0 卷的组件，而这些卷又是 RAID-1 卷的子镜像，请针对每个片运行 `metareplace -e` 命令。

此示例中使用的是 `/dev/dsk/c0t1d0s4` 和镜像 `d10`。

```
# metareplace -e d10 c0t1d0s4
```

11 如果在 RAID-0 卷上构建了软分区，请针对每个 RAID-0 卷运行 `metarecover` 命令。

在此示例中，在 RAID-0 卷 `d17` 上构建了软分区。

```
# metarecover d17 -m -p
```

12 替换已删除的热备件，然后将它们添加到适当的热备用池中。

在此示例中，热备用池 `hsp000` 包括 `c0t1d0s6`，该片随后添加到该热备用池中。

```
# metahs -a hsp000 c0t1d0s6hsp000: Hotspare is added
```

13 如果软分区或非冗余卷受到磁盘故障的影响，请从备份中恢复数据。如果只有冗余卷受到影响，请对数据进行验证。

检查所有卷上的用户和应用程序数据。可能必须运行应用程序级一致性检查器或者使用某种其他方法来检查数据。

从磁盘移动问题中恢复

本节介绍在 Solaris Volume Manager 环境中移动磁盘之后如何从意外的问题中恢复。

磁盘移动和设备 ID 概述

Solaris Volume Manager 使用与特定磁盘相关联的设备 ID 来跟踪在 Solaris Volume Manager 配置中使用的所有磁盘。当将磁盘移到另一个控制器时，或者 SCSI 目标编号发生变化时，Solaris Volume Manager 通常可正确地识别移动，并相应地更新所有相关的 Solaris Volume Manager 记录。此时无需系统管理员介入。在个别情况下，Solaris Volume Manager 无法完全更新这些记录并在系统引导时报告错误。

解决未命名设备错误

如果您添加新硬件或者移动硬件（例如，将一组磁盘从一个控制器移到另一个控制器），Solaris Volume Manager 会检查与所移动磁盘相关联的设备 ID，并相应地更新内部 Solaris Volume Manager 记录中的 *cntndn* 名称。如果无法更新这些记录，那么，在系统引导时，由 `svc:/system/mdmonitor` 服务产生的引导过程会向控制台报告一个错误：

```
Unable to resolve unnamed devices for volume management.
```

```
Please refer to the Solaris Volume Manager documentation,
```

```
Troubleshooting section, at http://docs.sun.com or from
```

```
your local copy.
```

该问题既不会导致数据丢失，也不会直接产生任何负面影响。此错误消息表明 Solaris Volume Manager 名称记录只进行了部分更新。`metastat` 命令的输出中显示了一些以前使用过的 *cntndn* 名称，还显示了一些反映移动后状态的 *cntndn* 名称。

在这种情况下，如果需要更新 Solaris Volume Manager 配置，则在发出任何 `meta*` 命令时，必须使用由 `metastat` 命令报告的 *cntndn* 名称。

如果出现此错误状态，可以执行以下操作之一来解决此问题：

- 将所有的磁盘都恢复到其原来的位置。接着，对重新配置执行重新引导，或者运行以下命令（作为单个命令）：

```
/usr/sbin/devfsadm && /usr/sbin/metadevadm -r
```

在完成这些命令之后，错误状态将得以解决。

- 请与技术支持代表联系以寻求帮助。

注 - 此错误状态极少出现。如果确实出现了，则很有可能会影响与光纤通道连接的存储器。

升级到 Solaris 10 发行版之后设备 ID 的改变

从 Solaris 10 发行版开始，设备 ID 的输出以新格式显示。Solaris Volume Manager 可以用新旧两种格式显示设备 ID 输出，具体情况取决于设备 ID 信息是何时添加到状态数据库副本中的。

以前，设备 ID 显示为十六进制值。新的格式将设备 ID 显示为 ASCII 字符串。在许多情况下，这种变化极其微小，如以下示例所示：

旧格式：`id1,ssd@w600c0ff00000000007ecd255a9336d00`

新格式：`id1,ssd@n600c0ff00000000007ecd255a9336d00`

在其他情况下，这种变化较为显著，如以下示例所示：

旧格式：`id1,sd@w4849544143484920444b3332454a2d33364e4320202020203433334239383939`

新格式：`id1,ssd@n600c0ff00000000007ecd255a9336d00`

在升级到 Solaris 10 发行版时，与以前的 Solaris 发行版中创建的现存磁盘集相关联的设备 ID 的格式，在 Solaris Volume Manager 配置中不会进行更新。如果需要恢复到以前的 Solaris 发行版，那么，在升级之后对磁盘集进行的配置更改在以前的发行版中可能会不可用。这些配置更改包括：

- 向升级之前存在的磁盘集内添加新磁盘
- 创建新磁盘集
- 创建状态数据库副本

这些配置更改可能会影响所有能够在 Solaris Volume Manager 中创建的磁盘集，包括本地集。例如，如果对 Solaris 10 发行版中创建的磁盘集进行了上述任何更改，就不能将该磁盘集导入到以前的 Solaris 发行版中。另一个示例是，您可能会将镜像根的一面升级到 Solaris 10 发行版，然后对本地集进行配置更改。如果随后将该子镜像又重新合并到以前的 Solaris 发行版中，系统将无法识别这些更改。

Solaris 10 OS 配置总是以新格式显示设备 ID，即便是在升级期间也是如此。可以使用 `prtconf -v` 命令来显示此信息。而 Solaris Volume Manager 会以旧格式或新格式显示，具体情况取决于在开始使用该磁盘时所运行的 Solaris OS 的版本。要确定 Solaris Volume Manager 所显示的设备 ID 是否与 Solaris OS 配置中的设备 ID 格式不同但等效，请将 `metastat` 命令的输出与 `prtconf -v` 命令的输出进行比较。

在以下示例中，对于同一个磁盘 `c1t6d0`，`metastat` 命令输出中显示的设备 ID 与 `prtconf -v` 命令输出中的设备 ID 格式不同但等效。

metastat

d127: Concat/Stripe

Size: 17629184 blocks (8.4 GB)

Stripe 0:

Device	Start Block	Dbase	Reloc
c1t6d0s2	32768	Yes	Yes

Device Relocation Information:

Device	Reloc	Device ID	c1t6d0	Yes	id1,sd@w4849544143484920444b3332454a2d33364e43202020203433334239383939
--------	-------	-----------	---------------	-----	---

prtconf -v

.(output truncated)

.
.

sd, **instance #6**

System properties:

name='lun' type=int items=1
value=00000000

name='target' type=int items=1
value=00000006

name='class' type=string items=1
value='scsi'

Driver properties:

name='pm-components' type=string items=3 dev=none

```

        value='NAME=spindle-motor' + '0=off' + '1=on'

name='pm-hardware-state' type=string items=1 dev=none

        value='needs-suspend-resume'

name='ddi-failfast-supported' type=boolean dev=none

name='ddi-kernel-ioctl' type=boolean dev=none

Hardware properties:

name='devid' type=string items=1

        value='id1,@THITACHI_DK32EJ-36NC____433B9899'

.

.

.

(output truncated)

```

`prtconf -v` 命令输出中包含 "instance #6" 的行与 `metastat` 命令输出中的磁盘 `c1t6d0` 相关联。 `prtconf -v` 命令输出中的设备 ID (`id1,@THITACHI_DK32EJ-36NC____433B9899`) 与 `metastat` 命令输出中的设备 ID (`id1,sd@w4849544143484920444b3332454a2d33364e43202020203433334239383939`) 相关联。这两个命令输出之间的区别表明，Solaris Volume Manager 在 `metastat` 命令的输出中以十六进制格式显示设备 ID，而 Solaris 10 OS 配置在 `prtconf` 命令的输出中以 ASCII 字符串的形式显示设备 ID。

从引导问题中恢复

由于 Solaris Volume Manager 允许您镜像根 (/)、`swap` 和 `/usr` 目录，因此在引导系统时可能会出现特殊问题。硬件故障或操作员错误都有可能引起这些问题。本节中的过程提供了针对类似潜在问题的解决方案。

下表描述了这些问题并介绍了相应的解决方案。

表 25-1 Solaris Volume Manager 的常见引导问题

出现引导问题的原因	参考
/etc/vfstab 文件中包含错误的信息。	第 300 页中的 “如何从错误的 /etc/vfstab 项中恢复”
未定义足够的状态数据库副本。	第 310 页中的 “如何从状态数据库副本不足中恢复”
引导设备（磁盘）出现故障。	第 303 页中的 “如何从引导设备故障中恢复”

有关引导问题的背景信息

- 如果 Solaris Volume Manager 由于错误而使卷脱机，请取消挂载出现故障的磁盘上的所有文件系统。
由于磁盘上的每个片都是相互独立的，因此可以在一个磁盘上挂载多个文件系统。如果软件遇到了故障，则同一个磁盘上的其他片将有可能很快遇到故障。直接在磁盘片上挂载的文件系统不具有 Solaris Volume Manager 错误处理保护功能。如果使这样的文件系统保持挂载，则很可能会出现系统崩溃和数据丢失的情况。
- 请尽可能缩短使用已禁用或处于脱机状态的子镜像运行的时间。在重新同步过程中和联机备份间隔期间，镜像将失去所有的保护。

如何从错误的 /etc/vfstab 项中恢复

如果在 /etc/vfstab 文件中创建了一个错误的项（例如，在镜像根 (/) 文件系统时），系统最初看起来在正确引导，但随后会失败。要解决此问题，需要在单用户模式下编辑 /etc/vfstab 文件。

下面是从 /etc/vfstab 文件中错误的项进行恢复所需的概括性步骤：

1. 在单用户模式下引导系统
2. 针对镜像卷运行 fsck 命令
3. 重新挂载文件系统并启用读写选项
4. 可选：针对根 (/) 镜像运行 metaroot 命令
5. 验证 /etc/vfstab 文件是否能够正确地引用与该文件系统项相对应的卷
6. 重新引导系统

▼ 从根 (/) RAID-1（镜像）卷中恢复

在以下示例中，根 (/) 文件系统是用双向镜像 d0 镜像的。/etc/vfstab 文件中的根 (/) 项由于某种原因而恢复到文件系统的原始片。但是，/etc/system 文件中的信息仍显示为从镜像 d0 引导。最有可能的原因就是，没有使用 metaroot 命令来维护 /etc/system 和 /etc/vfstab 文件。另一个可能的原因就是，将 /etc/vfstab 文件的旧副本又重新复制到当前的 /etc/vfstab 文件中。

错误的 /etc/vfstab 文件与以下信息类似：

```
#device      device      mount      FS      fsck  mount  mount
#to mount    to fsck      point      type    pass  at boot options
#
/dev/dsk/c0t3d0s0 /dev/rdisk/c0t3d0s0 /      ufs      1      no      -
/dev/dsk/c0t3d0s1 -              -      swap     -      no      -
/dev/dsk/c0t3d0s6 /dev/rdisk/c0t3d0s6 /usr    ufs      2      no      -
#
/proc        -              /proc    proc     -      no      -
swap         -              /tmp     tmpfs    -      yes     -
```

由于出现错误，您将在系统引导时自动进入单用户模式：

```
ok boot
...
configuring network interfaces: hme0.

Hostname: host1

mount: /dev/dsk/c0t3d0s0 is not this fstype.

setmnt: Cannot open /etc/mnttab for writing

INIT: Cannot create /var/adm/utmp or /var/adm/utmpx

INIT: failed write of utmpx entry:" "
```



```
INIT: failed write of utmpx entry:" "
```

INIT: SINGLE USER MODE

Type Ctrl-d to proceed with normal startup,

(or give root password for system maintenance): *<root-password>*

此时，根 (/) 和 /usr 文件系统都以只读方式挂载。请按照以下步骤操作：

1 针对根 (/) 镜像运行 fsck 命令。

注 - 注意使用根 (/) 镜像的正确的卷。

```
# fsck /dev/md/rdisk/d0
```

```
** /dev/md/rdisk/d0
```

```
** Currently Mounted on /
```

```
** Phase 1 - Check Blocks and Sizes
```

```
** Phase 2 - Check Pathnames
```

```
** Phase 3 - Check Connectivity
```

```
** Phase 4 - Check Reference Counts
```

```
** Phase 5 - Check Cyl groups
```

```
2274 files, 11815 used, 10302 free (158 frags, 1268 blocks,
```

```
0.7% fragmentation)
```

2 重新以读/写方式挂载根 (/) 文件系统，以便您可以编辑 /etc/vfstab 文件。

```
# mount -o rw,remount /dev/md/dsk/d0 /
```

```
mount: warning: cannot lock temp file </etc/.mnt.lock>
```

3 运行 metaroot 命令。

```
# metaroot d0
```

此命令会编辑 /etc/system 和 /etc/vfstab 文件，以便指定根 (/) 文件系统现在位于 d0 卷上。

- 4 验证 `/etc/vfstab` 文件是否包含正确的卷项。
`/etc/vfstab` 文件中的根 (`/`) 项应如下所示，从而使文件系统的项可以正确地引用 RAID-1 卷：

```
#device          device          mount   FS      fsck    mount    mount
#to mount        to fsck          point   type    pass    at boot  options

#
/dev/md/dsk/d0    /dev/md/rdisk/d0    /        ufs     1       no       -
/dev/dsk/c0t3d0s1 -                    -        swap    -       no       -
/dev/dsk/c0t3d0s6 /dev/rdisk/c0t3d0s6 /usr     ufs     2       no       -

#
/proc            -                /proc    proc    -       no       -
swap            -                /tmp     tmpfs   -       yes      -
```

- 5 重新引导系统。
系统将恢复到正常的运行状态。

▼ 如何从引导设备故障中恢复

如果您有一个根 (`/`) 镜像，那么，当引导设备出现故障时，您需要设置备用引导设备。

以下是此任务的概括性步骤：

- 从备用根 (`/`) 子镜像中引导
- 确定出现错误的状态数据库副本和卷
- 修理出现故障的磁盘
- 将状态数据库副本和卷恢复到其原始状态

最初，当引导设备出现故障时，您将看到一条与以下示例类似的消息。该消息可能会因体系结构而异。

```
Rebooting with command:

Boot device: /iommu/sbus/dma@f,81000/esp@f,80000/sd@3,0

The selected SCSI device is not responding

Can't open boot device
```

...

在看到此消息时，请记住该设备。然后按照以下步骤操作：

1 从另一个根 (/) 子镜像引导。

此示例中共有六个状态数据库副本，由于只有两个副本出现错误，因此仍可以引导系统。如果不是这种情况，则需要在单用户模式下删除不可访问的状态数据库副本。此过程在[第 310 页](#)中的“[如何从状态数据库副本不足中恢复](#)”中介绍。

在为根 (/) 文件系统创建镜像时，应在执行过程时记录备用根设备。在此示例中，disk2 是需要记录的备用根设备。

ok boot disk2

SunOS Release 5.9 Version s81_51 64-bit

Copyright 1983-2001 Sun Microsystems, Inc. All rights reserved.

Hostname: demo

...

demo console login: root

Password: <root-password>

Dec 16 12:22:09 host1 login: ROOT LOGIN /dev/console

Last login: Wed Dec 12 10:55:16 on console

Sun Microsystems Inc. SunOS 5.9 s81_51 May 2002

...

2 使用 metadb 命令来确定有多少个状态数据库副本失败。

metadb

flags		first blk	block count	
M	p	unknown	unknown	/dev/dsk/c0t3d0s3
M	p	unknown	unknown	/dev/dsk/c0t3d0s3
a m	p luo	16	1034	/dev/dsk/c0t2d0s3
a	p luo	1050	1034	/dev/dsk/c0t2d0s3


```
a      p      luo      16      1034      /dev/dsk/c0t1d0s3

a      p      luo      1050     1034      /dev/dsk/c0t1d0s3
```

在此示例中，系统无法再在 /dev/dsk/c0t3d0s3 片上检测状态数据库副本，该片位于出现故障的磁盘上。

3 使用 `metastat` 命令确定根 (/)、`swap` 和 `usr` 镜像的一半是否已失败。

```
# metastat

d0: Mirror

    Submirror 0: d10

        State: Needs maintenance

    Submirror 1: d20

        State: Okay

...

d10: Submirror of d0

    State: Needs maintenance

    Invoke: "metareplace d0 /dev/dsk/c0t3d0s0 <new device>"

    Size: 47628 blocks

    Stripe 0:

        Device              Start Block  Dbase State          Hot Spare

        /dev/dsk/c0t3d0s0      0          No    Maintenance
```

```
d20: Submirror of d0

    State: Okay

    Size: 47628 blocks

    Stripe 0:
```

```

Device          Start Block  Dbase State      Hot Spare

/dev/dsk/c0t2d0s0      0      No      Okay

d1: Mirror

  Submirror 0: d11

    State: Needs maintenance

  Submirror 1: d21

    State: Okay

...

d11: Submirror of d1

  State: Needs maintenance

  Invoke: "metareplace d1 /dev/dsk/c0t3d0s1 <new device>"

  Size: 69660 blocks

  Stripe 0:

    Device          Start Block  Dbase State      Hot Spare

    /dev/dsk/c0t3d0s1      0      No      Maintenance

d21: Submirror of d1

  State: Okay

  Size: 69660 blocks

  Stripe 0:

    Device          Start Block  Dbase State      Hot Spare

    /dev/dsk/c0t2d0s1      0      No      Okay
```

```
d2: Mirror

  Submirror 0: d12

    State: Needs maintenance

  Submirror 1: d22

    State: Okay

...

d12: Submirror of d2

  State: Needs maintenance

  Invoke: "metareplace d2 /dev/dsk/c0t3d0s6 <new device>"

  Size: 286740 blocks

  Stripe 0:

    Device                Start Block  Dbase State      Hot Spare

    /dev/dsk/c0t3d0s6      0           No    Maintenance

...

d22: Submirror of d2

  State: Okay

  Size: 286740 blocks

  Stripe 0:

    Device                Start Block  Dbase State      Hot Spare

    /dev/dsk/c0t2d0s6      0           No    Okay
```

在此示例中，metastat 命令显示以下子镜像需要维护：

- 位于设备 c0t3d0s0 上的子镜像 d10

- 位于设备 `c0t3d0s1` 上的子镜像 `d11`
 - 位于设备 `c0t3d0s6` 上的子镜像 `d12`
- 4 停止系统，然后替换磁盘。使用 `format` 命令或 `fmthard` 命令按照磁盘在出现故障之前的状态对磁盘进行分区。

提示 - 如果新磁盘与现有的磁盘相同（在此示例中为镜像中保持不变的一面），请快速格式化新磁盘。为此，请使用 `prtvtoc /dev/rdisk/c0t2d0s2 | fmthard -s - /dev/rdisk/c0t3d0s2` 命令（在此示例中为 `c0t3d0`）。

```
# halt

...

Halted

...

ok boot

...

# format /dev/rdisk/c0t3d0s0
```

- 5 重新引导系统。
- 请注意，必须从根 (/) 镜像的另一半重新引导。在创建镜像时，应当已经记录了备用根设备。

```
# halt

...

ok boot disk2
```

- 6 要删除失败的状态数据库副本并重新添加它们，请使用 `metadb` 命令。

```
# metadb
```

	flags	first blk	block count	
M	p	unknown	unknown	/dev/dsk/c0t3d0s3
M	p	unknown	unknown	/dev/dsk/c0t3d0s3
a m	p luo	16	1034	/dev/dsk/c0t2d0s3
a	p luo	1050	1034	/dev/dsk/c0t2d0s3

```
a      p      luo      16      1034      /dev/dsk/c0t1d0s3

a      p      luo      1050      1034      /dev/dsk/c0t1d0s3

# metadb -d c0t3d0s3

# metadb -c 2 -a c0t3d0s3

# metadb

      flags      first blk      block count

a m      p      luo      16      1034      /dev/dsk/c0t2d0s3

a      p      luo      1050      1034      /dev/dsk/c0t2d0s3

a      p      luo      16      1034      /dev/dsk/c0t1d0s3

a      p      luo      1050      1034      /dev/dsk/c0t1d0s3

a      u      16      1034      /dev/dsk/c0t3d0s3

a      u      1050      1034      /dev/dsk/c0t3d0s3
```

7 使用 metareplace 命令重新启用子镜像。

```
# metareplace -e d0 c0t3d0s0

Device /dev/dsk/c0t3d0s0 is enabled


# metareplace -e d1 c0t3d0s1

Device /dev/dsk/c0t3d0s1 is enabled


# metareplace -e d2 c0t3d0s6

Device /dev/dsk/c0t3d0s6 is enabled
```

在一段时间之后，重新同步将完成。现在可以返回到从原始设备引导。

从状态数据库副本故障中恢复

如果无法满足状态数据库副本定额（例如，由于驱动器故障），则系统无法重新引导到多用户模式下。当 Solaris Volume Manager 发现可用的状态数据库副本不足一半时，则会崩溃。如果系统是使用恰好一半或者少于一半的功能状态数据库副本重新引导的，也可能出现这种情况。在 Solaris Volume Manager 术语中，状态数据库已变得“过时”。此过程解释如何从该问题中恢复。

▼ 如何从状态数据库副本不足中恢复

- 1 引导系统。

- 2 确定哪些状态数据库副本不可用。

```
# metadb -i
```

- 3 如果已知有一个或多个磁盘不可用，请删除这些磁盘上的状态数据库副本。否则，请删除足够数量的错误的状态数据库副本（由 metadb 报告的标有 W、M、D、F 或 R 状态标志的数据库副本），以确保现有状态数据库副本中的大多数都没有错误。

```
# metadb -d disk-slice
```

提示 – 带有大写状态标志的状态数据库副本有错误，带有小写状态标志的状态数据库副本可以正常工作。

- 4 验证副本是否已删除。

```
# metadb
```

- 5 重新引导系统。

```
# reboot
```

- 6 如有必要，请替换磁盘，正确格式化该磁盘，然后添加该磁盘所需的任何状态数据库副本。

请按照第 70 页中的“创建状态数据库副本”中的说明操作。

在拥有替换磁盘之后，停止系统，替换出现故障的磁盘，然后再次重新引导系统。使用 format 命令或 fmthard 命令按照磁盘在出现故障之前的状态对磁盘进行分区。

示例 25-1 从过时的状态数据库副本中恢复

在以下示例中，包含七个副本的磁盘已经出现错误。因此，系统只有三个良好的副本。系统将崩溃，因而无法重新引导到多用户模式下。

```

panic[cpu0]/thread=70a41e00: md: state database problem

403238a8 md:mddb_commitrec_wrapper+6c (2, 1, 70a66ca0, 40323964, 70a66ca0, 3c)

    %l0-7: 0000000a 00000000 00000001 70bbcce0 70bbcd04 70995400 00000002 00000000
40323908 md:alloc_entry+c4 (70b00844, 1, 9, 0, 403239e4, ff00)

    %l0-7: 70b796a4 00000001 00000000 705064cc 70a66ca0 00000002 00000024 00000000
40323968 md:md_setdevname+2d4 (7003b988, 6, 0, 63, 70a71618, 10)

    %l0-7: 70a71620 00000000 705064cc 70b00844 00000010 00000000 00000000 00000000
403239f8 md:setnm_ioctl+134 (7003b968, 100003, 64, 0, 0, ffbffc00)

    %l0-7: 7003b988 00000000 70a71618 00000000 00000000 000225f0 00000000 00000000
40323a58 md:md_base_ioctl+9b4 (157ffff, 5605, ffbffa3c, 100003, 40323ba8, ff1b5470)

    %l0-7: ff3f2208 ff3f2138 ff3f26a0 00000000 00000000 00000064 ff1396e9 00000000
40323ad0 md:md_admin_ioctl+24 (157ffff, 5605, ffbffa3c, 100003, 40323ba8, 0)

    %l0-7: 00005605 ffbffa3c 00100003 0157ffff 0aa64245 00000000 7efefeff 81010100
40323b48 md:mdioctl+e4 (157ffff, 5605, ffbffa3c, 100003, 7016db60, 40323c7c)

    %l0-7: 0157ffff 00005605 ffbffa3c 00100003 0003ffff 70995598 70995570 0147c800
40323bb0 genunix:ioctl+ldc (3, 5605, ffbffa3c, ffffffff8, fffffffe0, ffbffa65)

    %l0-7: 0114c57c 70937428 ff3f26a0 00000000 00000001 ff3b10d4 0aa64245 00000000

panic:

stopped at      edd000d8:      ta      %icc,%g0 + 125

Type 'go' to resume

ok boot -s

```

Resetting ...

Sun Ultra 5/10 UPA/PCI (UltraSPARC-IIi 270MHz), No Keyboard

OpenBoot 3.11, 128 MB memory installed, Serial #9841776.

Ethernet address 8:0:20:96:2c:70, Host ID: 80962c70.

Rebooting with command: boot -s

Boot device: /pci@1f,0/pci@1,1/ide@3/disk@0,0:a File and args: -s

SunOS Release 5.9 Version s81_39 64-bit

Copyright 1983-2001 Sun Microsystems, Inc. All rights reserved.

configuring IPv4 interfaces: hme0.

Hostname: dodo

metainit: dodo: stale databases

Insufficient metadevice database replicas located.

Use metadb to delete databases which are broken.

Ignore any "Read-only file system" error messages.

Reboot the system when finished to reload the metadevice database.

After reboot, repair any broken database replicas which were deleted.

Type control-d to proceed with normal startup,
(or give root password for system maintenance): *root-password*
single-user privilege assigned to /dev/console.
Entering System Maintenance Mode

Jun 7 08:57:25 su: 'su root' succeeded for root on /dev/console

Sun Microsystems Inc. SunOS 5.9 s81_39 May 2002

metadb -i

flags		first blk	block count	
a	m p lu	16	8192	/dev/dsk/c0t0d0s7
a	p l	8208	8192	/dev/dsk/c0t0d0s7
a	p l	16400	8192	/dev/dsk/c0t0d0s7
M	p	16	unknown	/dev/dsk/c1t1d0s0
M	p	8208	unknown	/dev/dsk/c1t1d0s0
M	p	16400	unknown	/dev/dsk/c1t1d0s0
M	p	24592	unknown	/dev/dsk/c1t1d0s0
M	p	32784	unknown	/dev/dsk/c1t1d0s0
M	p	40976	unknown	/dev/dsk/c1t1d0s0
M	p	49168	unknown	/dev/dsk/c1t1d0s0

metadb -d c1t1d0s0

metadb

flags		first blk	block count	
a	m p lu	16	8192	/dev/dsk/c0t0d0s7

```

a p l      8208      8192      /dev/dsk/c0t0d0s7

a p l      16400     8192      /dev/dsk/c0t0d0s7

#
```

系统之所以会崩溃，是因为它无法再检查 `/dev/dsk/c1t1d0s0` 片上的状态数据库副本。该片位于出现故障的磁盘上或者该片已附加到出现故障的控制器。第一个 `metadb - i` 命令将该片上的副本标识为主块有问题。

当您删除过时的状态数据库副本时，根 (`/`) 文件系统是只读的。可以忽略所显示的 `mddb.cf` 错误消息。

此时，系统将重新正常工作，尽管它的状态数据库副本数目有可能比应有的少。如果卷使用已出现故障的存储器的一部分，则其状态同样为有故障的、有错误的或热备用的。这些问题应当立即解决。

从软分区问题中恢复

本节介绍如何恢复软分区的配置信息。只有当所有的状态数据库副本已丢失，而且没有下列信息之一时，才应当使用以下过程：

- `metastat -p` 输出的当前副本或准确副本
- `md.cf` 文件的当前副本或准确副本
- 最新的 `md.tab` 文件

▼ 如何恢复软分区的配置数据

在每个软分区扩展盘区的开头，有一个用于标记软分区扩展盘区的开头的扇区。这些隐藏的扇区称为**分区扩展表头**。软分区的用户看不到这些标头。如果所有的 Solaris Volume Manager 配置数据已丢失，则可以对磁盘进行扫描以尝试生成配置数据。

不到万不得已，不要使用此过程来恢复所丢失的软分区配置信息。只有当 `metadb` 和 `md.cf` 文件均丢失，而且 `md.tab` 文件已过时或者已丢失时，才应当使用 `metarecover` 命令。

注-此过程仅适用于恢复软分区信息，而不适用于从丢失其他配置中进行恢复或者恢复其他 Solaris Volume Manager 卷的配置信息。

注 – 如果配置中包括在软分区之上构建的其他 Solaris Volume Manager 卷，则应当在尝试恢复其他卷之前恢复软分区。

有关软分区的配置信息存储在设备上和状态数据库中。由于这两个源都有可能会损坏，因此必须向 `metarecover` 命令指示哪个源可靠。

首先，使用 `metarecover` 命令来确定这两个源是否一致。如果它们一致，就不能使用 `metarecover` 命令进行任何更改。但是，如果 `metarecover` 命令报告存在不一致，则必须认真检查它的输出，以确定磁盘或状态数据库是否已经损坏。然后，应当使用 `metarecover` 命令来基于适当的源重新构建配置。

- 1 请阅读第 154 页中的“软分区的配置指南”。
- 2 使用 `metarecover` 命令检查软分区的恢复信息。

```
# metarecover component -p -d
```

component 指定原始组件的 *cntndnsn* 名称

-p 指定要重新生成软分区

-d 指定要扫描软分区上分区扩展表头的物理片

示例 25-2 从磁盘上的分区扩展表头中恢复软分区

```
# metarecover c1t1d0s1 -p -d
```

The following soft partitions were found and will be added to
your metadvice configuration.

Name	Size	No. of Extents
d10	10240	1
d11	10240	1
d12	10240	1

```
# metarecover c1t1d0s1 -p -d
```

The following soft partitions were found and will be added to
your metadvice configuration.

Name	Size	No. of Extents
------	------	----------------

d10	10240	1
d11	10240	1
d12	10240	1

WARNING: You are about to add one or more soft partition
metadevices to your metadevice configuration. If there
appears to be an error in the soft partition(s) displayed
above, do NOT proceed with this recovery operation.
Are you sure you want to do this (yes/no)?**yes**
clt1d0s1: Soft Partitions recovered from device.

bash-2.05# metastat
d10: Soft Partition

Device: clt1d0s1
State: Okay
Size: 10240 blocks

Device	Start Block	Dbase Reloc
clt1d0s1	0	No Yes

Extent	Start Block	Block count
0	1	10240

d11: Soft Partition
Device: clt1d0s1
State: Okay
Size: 10240 blocks

Device	Start Block	Dbase Reloc
clt1d0s1	0	No Yes

Extent	Start Block	Block count
0	10242	10240

d12: Soft Partition

Device: clt1d0s1

State: Okay

Size: 10240 blocks

Device	Start Block	Dbase Reloc
clt1d0s1	0	No Yes

Extent	Start Block	Block count
0	20483	10240

在此示例中，在意外删除状态数据库副本之后，从磁盘中恢复三个软分区。

从另一个系统中恢复存储

甚至可以将 Solaris Volume Manager 配置从原始系统恢复到另一个系统上。

▼ 如何从本地磁盘集恢复存储

如果遇到系统故障，则可以将存储器连接到另一个系统，并从本地磁盘集恢复完整的配置。例如，假设系统中有一个由六个磁盘组成的外部磁盘组，某些磁盘上具有 Solaris Volume Manager 配置（至少包括一个状态数据库副本）。如果系统出现故障，则可以将这个磁盘组以物理方式移到新系统中，并使新系统能够识别该配置。此过程说明如何将磁盘移到另一个系统以及如何从本地磁盘集恢复配置。

注 - 此恢复过程仅适用于 Solaris 9 和更高版本的 Solaris Volume Manager 卷。

- 1 将包含 Solaris Volume Manager 配置的磁盘连接到原本没有 Solaris Volume Manager 配置的系统。

- 2 执行重新配置重新引导，以确保系统能够识别新添加的磁盘。

```
# reboot -- -r
```

- 3 确定新添加磁盘上包含状态数据库副本的片的主设备号/从设备号。

运行 `ls -lL`，并记下组名和日期之间的两个数字。这些数字就是该片的主设备号/从设备号。

```
# ls -lL /dev/dsk/clt9d0s7
```

```
brw-r----- 1 root sys 32, 71 Dec 5 10:05 /dev/dsk/clt9d0s7
```

- 4 如有必要，请通过在 `/etc/name_to_major` 中查找主设备号来确定与主设备号相对应的主设备名称。

```
# grep " 32" /etc/name_to_major sd 32
```

- 5 用指示 Solaris Volume Manager 在新磁盘上查找有效状态数据库副本的位置信息来更新 `/kernel/drv/md.conf` 文件。

例如，在以 `mddb_bootlist1` 开头的行中，用在步骤 4 中找到的主设备名称替换 `sd`。在此示例中，用在步骤 3 中标识的从设备号替换 `71`。

```
#pragma ident "@(#)md.conf 2.2 04/04/02 SMI"
```

```
#
```

```
# Copyright 2004 Sun Microsystems, Inc. All rights reserved.
```

```
# Use is subject to license terms.
```

```
#
```

```
# The parameters nmd and md_nsets are obsolete. The values for these
```

```
# parameters no longer have any meaning.
```

```
name="md" parent="pseudo" nmd=128 md_nsets=4;
```

```
# Begin MDD database info (do not edit)
```

```
mddb_bootlist1="sd:71:16:id0";
```

```
# End MDD database info (do not edit)
```

6 重新引导以强制 Solaris Volume Manager 重新装入配置。

将看到的消息与在控制台上显示的以下消息类似：

```
volume management starting.
```

```
Dec  5 10:11:53 host1 metadevadm: Disk movement detected
```

```
Dec  5 10:11:53 host1 metadevadm: Updating device names in
```

```
Solaris Volume Manager
```

```
The system is ready.
```

7 检验您的配置。使用 metadb 命令检验状态数据库副本的状态，使用 metastat 命令查看每个卷的状态。

```
# metadb
```

flags	first blk	block count	
a m p luo	16	8192	/dev/dsk/clt9d0s7
a luo	16	8192	/dev/dsk/clt10d0s7
a luo	16	8192	/dev/dsk/clt11d0s7
a luo	16	8192	/dev/dsk/clt12d0s7
a luo	16	8192	/dev/dsk/clt13d0s7

```
# metastat
```

```
d12: RAID
```

```
State: Okay
```

```
Interlace: 32 blocks
```

```
Size: 125685 blocks
```

```
Original device:
```

```
Size: 128576 blocks
```

Device	Start Block	Dbase	State	Reloc	Hot Spare
c1t11d0s3	330	No	Okay	Yes	
c1t12d0s3	330	No	Okay	Yes	
c1t13d0s3	330	No	Okay	Yes	

d20: Soft Partition

Device: d10
State: Okay
Size: 8192 blocks

Extent	Start Block	Block count
0	3592	8192

d21: Soft Partition

Device: d10
State: Okay
Size: 8192 blocks

Extent	Start Block	Block count
0	11785	8192

d22: Soft Partition

Device: d10
State: Okay
Size: 8192 blocks

Extent	Start Block	Block count
--------	-------------	-------------

0199788192

d10: Mirror

Submirror 0: d0

State: Okay

Submirror 1: d1

State: Okay

Pass: 1

Read option: roundrobin (default)

Write option: parallel (default)

Size: 82593 blocks

d0: Submirror of d10

State: Okay

Size: 118503 blocks

Stripe 0: (interlace: 32 blocks)

Device	Start Block	Dbase	State	Reloc	Hot Spare
c1t9d0s0	0	No	Okay	Yes	
c1t10d0s0	3591	No	Okay	Yes	

d1: Submirror of d10

State: Okay

Size: 82593 blocks

Stripe 0: (interlace: 32 blocks)

Device	Start Block	Dbase	State	Reloc	Hot Spare
c1t9d0s1	0	No	Okay	Yes	
c1t10d0s1	0	No	Okay	Yes	

Device Relocation Information:

Device	Reloc	Device ID
c1t9d0	Yes	id1,sd@SSEAGATE_ST39103LCSUN9.0GLS3487980000U00907AZ
c1t10d0	Yes	id1,sd@SSEAGATE_ST39103LCSUN9.0GLS3397070000W0090A8Q
c1t11d0	Yes	id1,sd@SSEAGATE_ST39103LCSUN9.0GLS3449660000U00904NZ
c1t12d0	Yes	id1,sd@SSEAGATE_ST39103LCSUN9.0GLS32655400007010H04J
c1t13d0	Yes	id1,sd@SSEAGATE_ST39103LCSUN9.0GLS3461190000701001T0

#

metadb

flags	first blk	block count	
a m p luo	16	8192	/dev/dsk/c1t9d0s7
a luo	16	8192	/dev/dsk/c1t10d0s7
a luo	16	8192	/dev/dsk/c1t11d0s7
a luo	16	8192	/dev/dsk/c1t12d0s7
a luo	16	8192	/dev/dsk/c1t13d0s7

metastat

d12: RAID

State: Okay

Interlace: 32 blocks

Size: 125685 blocks

Original device:

Size: 128576 blocks

Device	Start Block	Dbase	State	Reloc	Hot Spare
c1t11d0s3	330	No	Okay	Yes	
c1t12d0s3	330	No	Okay	Yes	
c1t13d0s3	330	No	Okay	Yes	

d20: Soft Partition

Device: d10

State: Okay

Size: 8192 blocks

Extent	Start Block	Block count
0	3592	8192

d21: Soft Partition

Device: d10

State: Okay

Size: 8192 blocks

Extent	Start Block	Block count
0	11785	8192

d22: Soft Partition

Device: d10

State: Okay

Size: 8192 blocks

Extent	Start Block	Block count
0	19978	8192

d10: Mirror

Submirror 0: d0

State: Okay

Submirror 1: d1

State: Okay

Pass: 1

Read option: roundrobin (default)

Write option: parallel (default)

Size: 82593 blocks

d0: Submirror of d10

State: Okay

Size: 118503 blocks

Stripe 0: (interlace: 32 blocks)

Device	Start Block	Dbase	State	Reloc	Hot Spare
c1t9d0s0	0	No	Okay	Yes	

c1t10d0s0	3591	No	Okay	Yes
-----------	------	----	------	-----

d1: Submirror of d10

State: Okay

Size: 82593 blocks

Stripe 0: (interlace: 32 blocks)

Device	Start Block	Dbase	State	Reloc	Hot Spare
c1t9d0s1	0	No	Okay	Yes	
c1t10d0s1	0	No	Okay	Yes	

Device Relocation Information:

Device	Reloc	Device ID
c1t9d0	Yes	id1,sd@SSEAGATE_ST39103LCSUN9.0GLS3487980000U00907AZ1
c1t10d0	Yes	id1,sd@SSEAGATE_ST39103LCSUN9.0GLS3397070000W0090A8Q
c1t11d0	Yes	id1,sd@SSEAGATE_ST39103LCSUN9.0GLS3449660000U00904NZ
c1t12d0	Yes	id1,sd@SSEAGATE_ST39103LCSUN9.0GLS32655400007010H04J
c1t13d0	Yes	id1,sd@SSEAGATE_ST39103LCSUN9.0GLS3461190000701001T0

metastat -p

d12 -r c1t11d0s3 c1t12d0s3 c1t13d0s3 -k -i 32b

d20 -p d10 -o 3592 -b 8192

d21 -p d10 -o 11785 -b 8192

```
d22 -p d10 -o 19978 -b 8192

d10 -m d0 d1 1

d0 1 2 c1t9d0s0 c1t10d0s0 -i 32b

d1 1 2 c1t9d0s1 c1t10d0s1 -i 32b

#
```

从已知的磁盘集恢复存储

在 Solaris Volume Manager 中引入了对磁盘集设备 ID 的支持，这允许您从已知的磁盘集恢复存储并将该磁盘集导入到其他系统。使用 `metainport` 命令，可以将已知的磁盘集从一个系统导入到另一个系统。这两个系统都必须包含现有的 Solaris Volume Manager 配置（包括设备 ID 支持）。有关设备 ID 支持的更多信息，请参见第 205 页中的“[磁盘集中的异步共享存储](#)”。有关 `metainport` 命令的更多信息，请参见 `metainport(1M)` 手册页。

▼ 如何列显有关可导入的磁盘集的报告

- 1 成为超级用户。
- 2 获取有关要导入的磁盘集的报告。

```
# metainport -r -v
```

-r 提供可导入到系统上的未配置磁盘集的报告。

-v 提供有关以下内容的详细信息：状态数据库副本的位置；可导入到系统上的未配置磁盘集的磁盘状态。

示例 25-3 报告可导入的磁盘集

以下示例说明如何列显有关可导入的磁盘集的报告。

```
# metainport -r
```

```
Drives in regular diskset including disk c1t2d0:
```

```
c1t2d0
```

```
c1t3d0
```

```
More info:
```

```

    metaimport -r -v c1t2d0

Import:  metaimport -s <newsetname> c1t2d0

Drives in replicated diskset including disk c1t4d0:

    c1t4d0

    c1t5d0

More info:

    metaimport -r -v c1t4d0

Import:  metaimport -s <newsetname> c1t4d0

```

metaimport -r -v c1t2d0

```

Import: metaimport -s <newsetname> c1t2d0

Last update: Mon Dec 29 14:13:35 2003

Device      offset      length replica flags
c1t2d0       16          8192      a      u
c1t3d0       16          8192      a      u
c1t8d0       16          8192      a      u

```

▼ 如何将磁盘集从一个系统导入到另一个系统

- 1 成为超级用户。
- 2 验证磁盘集是否可以导入。

```
# metaimport -r -v
```

- 3 导入可用的磁盘集。

```
# metaimport -s diskset-name drive-name
```

-s *diskset-name* 指定所创建的磁盘集的名称。

drive-name 标识正在导入的磁盘集中包含状态数据库副本的磁盘 (c#t#d#)。

4 验证该磁盘集是否已导入。

```
# metaset -s diskset-name
```

示例 25-4 导入磁盘集

以下示例说明如何导入磁盘集。

```
# metainport -s red c1t2d0

Drives in diskset including disk c1t2d0:

    c1t2d0

    c1t3d0

    c1t8d0

More info:  metainport -r -v c1t2d0# metaset -s red
```

Set name = red, Set number = 1

Host	Owner
host1	Yes

Drive	Dbase
c1t2d0	Yes
c1t3d0	Yes


```
clt8d0 Yes
```

从磁盘集问题中恢复

本节详述如何从与特定的磁盘集相关的问题中恢复。

无法获取磁盘集拥有权时应执行的操作

如果无法从任何节点获取磁盘集拥有权（可能由于系统故障、磁盘故障或通信链路故障），从而无法删除磁盘集记录，则可以从当前主机上的 Solaris Volume Manager 状态数据库副本记录中清除磁盘集。

清除磁盘集记录不会影响包含在磁盘集中的状态数据库信息，因此，可以在以后导入磁盘集（使用 `metainport` 命令，如第 222 页中的“导入磁盘集”中所述）。

如果需要从 Sun Cluster 配置中清除磁盘集，请使用以下过程，但是，在没有 Sun Cluster 配置时，应使用 `-C` 选项，而不要使用 `-P` 选项。

▼ 如何清除磁盘集

- 1 尝试使用 `metaset` 命令来获取磁盘集。

```
# metaset -s setname -t -f
```

此命令会尝试强制 (`-f`) 获取 (`-t`) 名为 `setname` 的磁盘集。如果可以获取该磁盘集，则此命令将成功。如果在运行此命令时，该磁盘集由另一个主机拥有，则另一个主机将崩溃，以防止数据损坏或丢失。如果此命令成功，则可以彻底删除该磁盘集，而无需清除它。

如果无法获取该磁盘集，则可以清除拥有权记录。

- 2 使用带 `-P` 的 `metaset` 命令从当前的主机中清除磁盘集。

```
# metaset -s setname -P
```

此命令将从运行该命令的主机中清除 (`-P`) 名为 `setname` 的磁盘集。

- 3 使用 `metaset` 命令验证该磁盘集是否已清除。

```
# metaset
```

示例 25-5 清除磁盘集

```
host1# metaset -s red -t -f

metaset: host1: setname "red": no such set

host2# metaset

Set name = red, Set number = 1

Host                Owner

host2

Drive   Dbase

c1t2d0  Yes

c1t3d0  Yes

c1t8d0  Yes

host2# metaset -s red -P

host2# metaset
```

另请参见

- [第 18 章](#)（介绍有关磁盘集的概念信息）。
- [第 19 章](#)（介绍有关与磁盘集相关联的任务的信息）。

使用 ufsdump 命令对已挂载的文件系统执行备份

以下过程说明在使用 ufsdump 命令来备份 RAID-1 卷上的已挂载文件系统时，如何提高该命令的性能。

▼ 如何针对 RAID-1 卷上的已挂载文件系统执行备份

可以使用 ufsdump 命令来备份 RAID-1 卷上已挂载文件系统的文件。当备份实用程序是 ufsdump 时，请将该卷上的读取策略设置为“first（优先）”。这样可以提高备份的执行速率。

- 1 成为超级用户。
- 2 运行 metastat 命令以确保镜像处于“Okay（正常）”状态。

```
# metastat d40
```

```
d40: Mirror
```

```
    Submirror 0: d41
```

```
        State: Okay
```

```
    Submirror 1: d42
```

```
        State: Okay
```

```
Pass: 1
```

```
Read option: roundrobin (default)
```

```
Write option: parallel (default)
```

```
Size: 20484288 blocks (9.8 GB)
```

应首先修复处于“Maintenance（维护）”状态的镜像。

- 3 将镜像的读取策略设置为“first（优先）”。

```
# metaparam -r first d40
```

```
# metastat d40
```

```
d40: Mirror
```

```
    Submirror 0: d41
```

```
State: Okay

Submirror 1: d42

State: Okay

Pass: 1

Read option: first

Write option: parallel (default)

Size: 20484288 blocks (9.8 GB)
```

4 对文件系统执行备份。

```
# ufsdump 0f /dev/backup /opt/test
```

5 在 ufsdump 命令完成之后，将镜像的读取策略设置为 "roundrobin"。

```
# metaparam -r roundrobin d40
```

```
# metastat d40
```

```
d40: Mirror

Submirror 0: d41

State: Okay

Submirror 1: d42

State: Okay

Pass: 1

Read option: roundrobin

Write option: parallel (default)

Size: 20484288 blocks (9.8 GB)
```

执行系统恢复

有时，从 DVD 或 CD 介质上的 Solaris OS 安装映像引导以执行系统恢复非常有用，重置 root 口令就是使用安装映像非常有用的一个示例。

如果使用的是 Solaris Volume Manager 配置，则需要挂载 Solaris Volume Manager 卷（而非基础磁盘）。如果对根 (/) 文件系统进行了镜像，则此步骤尤其重要。由于 Solaris Volume Manager 是 Solaris OS 的一部分，因此挂载 Solaris Volume Manager 卷将确保所做的更改在镜像的两面都反映出来。

使用以下过程，可以使 Solaris Volume Manager 卷能够从 Solaris OS DVD 或 CD-ROM 安装映像访问。

▼ 如何使用 Solaris Volume Manager 配置来恢复系统

从 Solaris OS 安装 DVD 或 CD 介质引导系统。从 Solaris miniroot 的 root 提示符执行此过程。

- 1 以只读方式挂载包含 Solaris Volume Manager 配置的基础磁盘。

```
# mount -o ro /dev/dsk/c0t0d0s0 /a
```

- 2 将 md.conf 文件复制到 /kernel/drv 目录中。

```
# cp /a/kernel/drv/md.conf /kernel/drv/md.conf
```

- 3 从 miniroot 挂载文件系统。

```
# umount /a
```

- 4 更新 Solaris Volume Manager 驱动程序以装入该配置。忽略由 update_drv 命令列显的任何警告消息。

```
# update_drv -f md
```

- 5 配置系统卷。

```
# metainit -r
```

- 6 如果在 Solaris Volume Manager 配置中有多个 RAID-1 卷，请将其重新同步。

```
# metasync mirror-name
```

- 7 Solaris Volume Manager 卷应当能够使用 mount 命令来访问。

```
# mount /dev/md/dsk/volume-name /a
```

示例 25-6 使用 Solaris Volume Manager 配置恢复系统

```
# mount -o ro /dev/dsk/c0t0d0s0 /a

# cp /a/kernel/drv/md.conf /kernel/drv/md.conf

# umount /a

# update_drv -f md

Cannot unload module: md

Will be unloaded upon reboot.

Forcing update of md.conf.

devfsadm: mkdir failed for /dev 0xled: Read-only file system

devfsadm: inst_sync failed for /etc/path_to_inst.1359: Read-only file system

devfsadm: WARNING: failed to update /etc/path_to_inst

# metainit -r

# metasync d0

# mount /dev/md/dsk/d0 /a
```

重要的 Solaris Volume Manager 文件

本附录包含有关 Solaris Volume Manager 文件的信息，以供参考。本附录包含以下主题：

- 第 335 页中的“系统文件和启动文件”
- 第 336 页中的“手动配置的文件”

系统文件和启动文件

本节介绍了 Solaris Volume Manager 正常运行所必需的文件。除了几项专门的配置更改以外，无需访问或修改这些文件。

- `/etc/lvm/mddb.cf`



注意 – 请勿编辑此文件。如果更改了此文件，可能会损坏 Solaris Volume Manager 配置。

`/etc/lvm/mddb.cf` 文件记录了状态数据库副本的位置。如果状态数据库副本位置发生更改，Solaris Volume Manager 将在 `mddb.cf` 文件中生成一项，用于记录所有状态数据库的位置。有关更多信息，请参见 `mddb.cf(4)` 手册页。

- `/etc/lvm/md.cf`

`/etc/lvm/md.cf` 文件包含为缺省（未指定或本地）磁盘集自动生成的配置信息。更改 Solaris Volume Manager 配置时，Solaris Volume Manager 会自动更新 `md.cf` 文件（正在使用的热备件的相关信息除外）。有关更多信息，请参见 `md.cf(4)` 手册页。



注意 – 请勿编辑此文件。如果更改了此文件，可能会损坏或无法恢复 Solaris Volume Manager 配置。

如果系统丢失了状态数据库中维护的信息，但只要未同时更改或创建卷，即可使用 `md.cf` 文件来恢复配置。请参见第 238 页中的“[如何从配置文件中初始化 Solaris Volume Manager](#)”。

- `/kernel/drv/md.conf`

Solaris Volume Manager 在启动时会读取 `md.conf` 配置文件。`md.conf` 文件包含状态数据库副本配置信息。从 Solaris 10 开始，不再手动编辑 `nmd` 和 `md_nsets` 参数。Solaris Volume Manager 已增强，可以根据需要动态配置卷。

手动配置的文件

md.tab 文件概述

`/etc/lvm/md.tab` 文件包含可用于重新构造 Solaris Volume Manager 配置的 Solaris Volume Manager 配置信息。Solaris Volume Manager 可以将此文件用作命令行实用程序 `metainit`、`metadb` 和 `metahs` 的输入，从而重新构造配置。卷、磁盘集和热备用池在此文件中可能有对应的项。有关创建此文件（通过使用 `metastat -p > /etc/lvm/md.tab` 命令）的说明，请参见第 238 页中的“[如何创建配置文件](#)”。

注 – `/etc/lvm/md.tab` 文件中的配置信息可能不同于正在使用的当前卷、热备件和状态数据库副本。系统管理员以手动方式使用此文件来捕获所需的配置。更改 Solaris Volume Manager 配置后，请重新创建此文件并保留备份副本。

创建并更新此文件后，`metainit`、`metahs` 和 `metadb` 命令随后便会激活在此文件中定义的卷、热备用池和状态数据库副本。

在 `/etc/lvm/md.tab` 文件中，使用 `metainit`、`metadb` 和 `metahs` 命令语法的每一行上都会显示单个卷的一个完整配置项。

注 – 如果使用 `metainit -an` 命令模拟初始化 `md.tab` 文件中的所有卷，则可能会显示与 `md.tab` 中定义的其他卷具有相关性的卷的错误消息。出现这些错误消息是由于 Solaris Volume Manager 没有维护运行 `metainit -an` 时应创建的卷的状态。如果配置存在，系统会根据现有配置对每一行进行评估。因此，即使表明 `metainit -an` 命令可能会失败，在不使用 `-n` 选项的情况下该命令也可能会成功。

然后，可以运行带有 `-a` 选项的 `metainit` 命令来激活 `/etc/lvm/md.tab` 文件中的所有卷，也可以在该命令中使用与此文件中的特定项对应的卷名来激活卷。

注 – Solaris Volume Manager 不会将配置信息写入或存储在 `/etc/lvm/md.tab` 文件中。必须手动编辑此文件并运行 `metainit`、`metabs` 或 `metadb` 命令才能创建 Solaris Volume Manager 组件。

有关更多信息，请参见 `md.tab(4)` 手册页。

Solaris Volume Manager 快速参考

本附录提供有关在 Solaris Volume Manager 中可用的特征和功能的快速访问信息。

命令行参考

此处列出了用于管理 Solaris Volume Manager 的所有命令。有关更多详细信息，请参见手册页。

表 B-1 Solaris Volume Manager 命令

Solaris Volume Manager 命令	说明	手册页
growfs	以非破坏性方式扩展 UFS 文件系统。	growfs(1M)
metaclear	删除活动的卷和热备用池。	metaclear(1M)
metadb	创建和删除状态数据库副本。	metadb(1M)
metadetach	将卷与 RAID-0 或 RAID-1（镜像）卷分离，或者将日志设备与事务卷分离。 注-不再支持事务卷。	metadetach(1M)
metadevadm	检查设备 ID 配置。	metadevadm(1M)
metahs	管理热备件和热备用池。	metahs(1M)
metaimport	将磁盘集（包括复制的磁盘集）导入在磁盘集中具备设备 ID 支持的现有 Solaris Volume Manager 配置。	metaimport(1M)
metainit	配置卷。	metainit(1M)
metaoffline	使子镜像脱机。	metaoffline(1M)
metaonline	使子镜像联机。	metaonline(1M)

表 B-1 Solaris Volume Manager 命令 (续)

Solaris Volume Manager 命令	说明	手册页
metaparam	修改卷参数。	metaparam(1M)
metarecover	恢复软分区的配置信息。	metarecover(1M)
metarename	重命名和交换卷名称。	metarename(1M)
metareplace	替换子镜像和 RAID-5 卷中的组件。	metareplace(1M)
metaroot	为镜像根 (/) 文件系统而设置系统文件。	metaroot(1M)
metaset	管理磁盘集。	metaset(1M)
metastat	显示卷或热备用池的状态。	metastat(1M)
metasync	在重新引导期间重新同步卷。	metasync(1M)
metattach	将组件连接到 RAID-0 或 RAID-1 卷。	metattach(1M)

Solaris Volume Manager CIM/WBEM API

管理 Solaris Volume Manager

Solaris Volume Manager 的 CIM/WBEM 应用程序编程接口 (Application Programming Interface, API) 提供了用于观察和配置 Solaris Volume Manager 的基于标准的公共程序接口。此 API 基于分布式管理任务组 (Distributed Management Task Force, DMTF) 的通用信息模型 (Common Information Model, CIM)。有关 DMTF 的更多信息，请参见 <http://www.dmtf.org>。

CIM 定义了称为“架构”的数据模型，该模型介绍以下信息：

- Solaris Volume Manager 设备的属性以及针对这些设备的操作
- 各种 Solaris Volume Manager 设备间的关系
- Solaris Volume Manager 设备和操作系统的其他方面（如文件系统）之间的关系

此模型可用于 Solaris 的基于 Web 的企业管理 (Web Based Enterprise Management, WBEM) SDK。WBEM SDK 是一组基于 Java™ 技术的 API，可用于访问 CIM 表示的系统管理功能。

有关 CIM/WBEM SDK 的更多信息，请参见《Solaris WBEM Developer's Guide》。

索引

C

cron 命令, 281

D

DiskSuite 工具, [请参见图形界面](#)

E

/etc/lvm/md.cf 文件, 335
/etc/lvm/mddb.cf 文件, 335
/etc/vfstab 文件, 145
 从错误的项中恢复, 300

F

fmthard 命令, 308, 310
format 命令, 308, 310

G

growfs 命令, 240-242, 339
growfs 命令, 41, 242
GUI, 样例, 36

I

I/O, 33

K

/kernel/drv/md.conf 文件, 336

L

lockfs 命令, 150

M

md.cf 文件, 336
 恢复 Solaris Volume Manager 配置, 238
md.tab 文件, 238
 概述, 336-337
mdmonitord 命令, 276-277
metaclear 命令, 339
metaclear 命令, 91, 142, 145-148
metadb 命令, 339
metadb 命令, 73
metadb 命令, 替换出现故障的磁盘和, 293
metadetach 命令, 339
metadetach 命令, 130, 142
metadevadm 命令, 339
 替换出现故障的磁盘和, 293
metahs 命令, 339
 启用, 193-194
 删除热备件, 192-193
 替换出现故障的磁盘和, 293
 替换热备件, 191-192
 向热备用池中添加片, 185-186
metainport 命令, 199-200, 222-225, 326-329, 339

metainit 命令, 339
metainit 命令, 239
metainit 命令, 创建热备用池, 184-185
metaoffline 命令, 339
metaoffline 命令, 131
metaonline 命令, 339
metaparam 命令, 340
metaparam 命令, 136
metaparam 命令
 更改热备用池的关联, 188-190
 将热备用池与卷相关联, 186-190
metarecover 命令, 340
 替换出现故障的磁盘和, 293
metarename 命令, 236, 340
metarename 命令, 237
metareplace 命令, 340
metareplace 命令, 132, 174, 309
metareplace 命令, 替换出现故障的磁盘和, 293
metaroot 命令, 340
metaset 命令, 340
 创建磁盘集, 208-209
 从磁盘集内删除磁盘, 216-217
 从磁盘集内删除主机, 220-222
 检查磁盘集的状态, 215-216
 删除磁盘集, 220-222
 释放磁盘集, 219-220
 提取磁盘集, 217-219
 向磁盘集内添加磁盘, 210
 向磁盘集内添加主机, 211-212
metastat 命令, 340
metastat 命令, 135, 172
metasync 命令, 340
metattach
 任务, 109, 113, 120, 124
metattach 命令, 340
metattach 命令, 89, 129, 138
 附加 RAID-5 组件, 173
 附加子镜像, 240

O

Oracle Real Application Clusters, 49-50

P

prtconf 命令, 查看设备 ID, 297-299

R

RAID, Solaris Volume Manager 中支持的级别, 30

RAID-0+1, 94-95

RAID-0 卷

 定义, 75

 使用情况, 75

RAID-0 (条带化) 卷, 包含三个片的示例, 76

RAID-1+0, 94-95

RAID-1 卷, 93-95

 maintenance (维护) 与 last erred (最近出错)
 , 244-245

 RAID-0+1, 94-95

 RAID-1+0, 94-95

 创建, 104-127

 读取和写入策略, 99

 镜像, 93-95

 替换和启用组件, 242-245

 选项, 98-99

 引导到单用户模式, 101

 有关替换和启用组件的信息, 245

 传送号, 99

 子镜像, 93-95

RAID-5 卷

 maintenance (维护) 与 last erred (最近出错)
 , 244-245

 包含扩展设备的示例, 163

 包含四张磁盘的示例, 162

 创建, 170-171

 定义, 30, 40

 概述, 161-164

 和交错, 164

 检查状态, 171-172

 扩展, 172-173, 173

 片状态, 165-166

 奇偶校验计算, 164

 奇偶校验信息, 161, 163

 启用出现故障的片, 174

 启用组件, 174

 任务, 169

 替换出现故障的片, 177

RAID-5 卷 (续)

- 替换和启用组件, 167, 242-245
- 替换组件, 174-177
- 要求, 164-165
- 有关替换和启用组件的信息, 245
- 指南, 164
- 重新同步片, 161
- 状态, 165-166

RAID-0 卷

- 类型, 75
- 指南, 81

RAID-0 (串联) 卷, 创建信息, 81**RAID-0 (串联条带化) 卷**

- 包含三个条带的示例, 79
- 交错值, 79

RAID-0 (条带化) 卷, 76

- 创建信息, 81

S**SCSI 磁盘**

- 替换, 293, 295, 296

SMF

- Solaris Volume Manager 服务, 46-47
- 升级 Solaris Volume Manager, 46-47

Solaris Volume Manager

- 请参见 Solaris Volume Manager
- Oracle Real Applications Clusters, 49-50
- Sun Cluster, 49-50
- 恢复配置, 238
- 配置原则, 44
- 网络存储设备, 249

Solaris Volume Manager for Sun Cluster

- 超时, 54-55
- 定向镜像读取, 55-57
- 基于应用程序的恢复, 55-57
- 配置, 54-55
- 软件组件, 50
- 数据管理和恢复, 55-57
- 优化的重新同步, 55-57

Solaris Volume Manager 界面

- Solaris Management Console, 36
- 命令行, 36
- 样例 GUI, 36

Solaris Volume Manager 元素, 概述, 38

Sun Cluster, 49-50

svc-mdmonitor 脚本, 276-277

swap, 取消镜像, 148

V

/var/adm/messages 文件, 243

替换出现故障的磁盘, 293

安

安装程序, 系统恢复, 333-334

保

保留磁盘集, 198-199

本

本地磁盘集, 195-196

并

并行访问, 定义, 76

并行写入策略, 99

查

查看 Solaris Volume Manager 配置, 228-235

常

常规性能指南, 32

串

串联

包含三个片的示例, 78-79

创建, 86

定义

另请参见RAID-0（串联）卷

扩展, 89

扩展 UFS 文件系统, 78

删除, 91

使用情况, 78

串联的条带, 删除, 91

串联卷, 请参见RAID-0（串联）卷

串联条带化

请参见RAID-0（串联条带化）卷

定义

另请参见RAID-0（串联条带化）卷

串行写入策略, 99

创

创建磁盘集, 208-209

创建配置文件, 238-240

磁

磁盘集, 195-196

包含两个共享磁盘集的示例, 203

保留, 198-199, 218

本地, 195-196

创建, 208-209

创建组件, 212-214

从磁盘集内删除主机, 220-222

导入, 199-200, 222-225

定义, 39, 43-44

多属主, 195-196

请参见多属主磁盘集

方案, 205

共享, 195-196

管理, 198-204

检查状态, 215-216

命名约定, 203

删除磁盘, 216-217

删除磁盘集, 220-222

磁盘集（续）

使用情况, 195-196

释放, 199, 219-220

提取, 217-219

添加磁盘, 210

添加主机, 211-212

已命名, 195-196

异步共享存储, 205

指南, 204

自动磁盘分区, 200-203

自动获取, 195-196

最大支持数目, 240

从

从安装程序中恢复系统, 333-334

从磁盘集内删除磁盘, 216-217

从磁盘集内删除主机, 220-222

错

错误, 使用脚本检查, 281-290

错误检查, 276-277

导

导入磁盘集, 199-200

定

定向镜像读取, 55-57

对

对磁盘重新分区, 200-203

多

多属主磁盘集, 195-196

多属主磁盘集 (续)

- RAID-1 卷, 55-57
- 导入, 49-50
- 任务, 53-54
- 设备 ID 支持, 49-50
- 已定义, 49-50
- 主节点, 51-54
- 多数一致算法, 63

服

服务管理工具 (Service Management Facility, SMF), 请参见SMF

副

副本, 43

更

更改缺省值, 240

共

共享的磁盘集, 195-196

故

故障转移配置, 43

管

管理磁盘集, 198-204

几

几何读取策略, 99

基

基于应用程序的恢复, 55-57

检

检查磁盘集的状态, 215-216

简

简单卷, 定义, 39

交

交错, 指定, 85
交错值, 76
交换卷名称, 236

界

界面, 请参见Solaris Volume Manager 界面

镜

镜像

请参见RAID-1 卷
另请参见RAID-1 卷
root (/) \ /usr 和 swap, 111
包含两个子镜像的示例, 94
创建, 104-127
定义, 39
读取和写入性能, 32
附加子镜像, 129
根文件系统
 SPARC, 112-116
 x86, 116-127
 使用 GRUB, 116-127
更改选项, 137
更新大小, 137-138
和联机备份, 149
可以取消挂载的文件系统, 109

镜像（续）

- 扩展, 138
- 双向镜像, 258-259, 259, 262-267, 267, 270-271, 271-272, 272-273
- 替换和启用组件, 167
- 文件系统, 107-111
- 样例状态输出, 133
- 重新同步, 96

卷

- 定义, 38
- 概念性概述, 39
- 交换名称, 236
- 扩展磁盘空间, 41-42
- 类型, 39
- 使用, 40
- 使用文件系统命令, 40
- 虚拟磁盘, 36
- 重命名, 237
- 最大支持数目, 240
- 卷名称切换, 42

配

- 配置, 查看, 228-235
- 配置规划
 - 概述, 30
 - 权衡因素, 32
 - 指南, 30
- 配置文件, 创建, 238-240

片

- 片, 扩展, 88

启

- 启用子镜像中的片, 132

热

热备件

- 定义, 179
- 概念性概述, 179-180
- 启用, 193-194
- 热备件如何用于 Solaris Volume Manager, 180
- 替换一个热备用池中的热备件, 192
- 添加到热备用池中, 186

热备用池, 43

- 包含镜像的示例, 181
- 创建, 184-185
- 定义, 39, 43, 180
- 概念性概述, 179-182
- 更改关联, 188-190
- 关联, 188
- 基本操作, 43
- 检查状态, 190-194
- 将片添加到, 185-186
- 删除热备件, 192-193
- 替换热备件, 191-192
- 维护, 190-194
- 与卷相关联, 186-190
- 状态, 181

软

- 软分区, 153
 - 创建, 155-156
 - 定义, 153
 - 恢复配置, 314
 - 检查状态, 157-160
 - 扩展, 158-159
 - 任务, 155
 - 删除, 159-160
 - 维护, 157-160
 - 位置, 153
 - 指南, 154

删

- 删除磁盘集, 220-222

设

设备 ID, 格式, 297-299

释

释放磁盘集, 199, 219-220

首

首先读取策略, 99

顺

顺序 I/O, 33

随

随机 I/O, 33-34

提

提取磁盘集, 217-219

添

添加热备件, 186

条

条带

创建, 85

定义

另请参见RAID-0（条带化）卷

扩展, 89

删除, 91

条带化, 定义, 76

条带化卷, 请参见RAID-0（条带化）卷

图

图形界面, 概述, 36

文

文件系统

扩展, 240-242

扩展概述, 41

取消镜像, 148

通过创建串联来扩展, 88

原则, 44

系

系统文件, 335-336

向

向磁盘集内添加磁盘, 210

向磁盘集内添加主机, 211-212

小

小型服务器, 部署概述, 247-248

循

循环读取策略, 99

疑

疑难解答

/etc/vfstab 文件中的错误项, 300

metaimport 命令, 326-329

从磁盘移动问题中恢复, 296

导入磁盘集, 326-329

设备 ID 的差异, 297-299

替换出现故障的磁盘, 293

一般原则, 292-293

疑难解答（续）

引导问题, 299-309

已

已命名的磁盘集, 195-196

引

引导到单用户模式, 101

引导问题, 299-309

元

元设备, [请参见卷](#)

在

在 RAID-5 卷中启用片, 174

在磁盘集内创建组件, 212-214

增

增强的存储, [请参见图形界面](#)

重

重命名卷, 235

重新同步

部分, 96

完整, 96

已优化, 96

主

主节点, 51-54

传

传送号

和只读镜像, 96

已定义, 99

状

状态, 216

状态数据库

定义, 39, 42

概念性概述, 42-43, 64

损坏, 63

状态数据库副本, 43

磁盘集的条件, 200-203

错误, 66

大小, 200-203

定义, 43

基本操作, 63

使用情况, 63

双磁盘配置, 66-67

添加更大的副本, 74

位置, 43, 65-66

在单个片上创建多个, 65-66

最小数目, 65-66

子

子镜像, 93-94

[请参见RAID-1 卷](#)

拆离, 94

附加, 94

启用出现故障的片, 132

替换出现故障的片, 140

替换整个, 142

在脱机时操作, 94

自

自动磁盘分区, 200-203

自动获取磁盘集, 195-196

自上而下创建卷

RAID 1 卷, 创建, 257-260

自上而下创建卷（续）

shell 脚本, 261-272

卷配置文件, 269

缺省值, 272-273

组

组件, 定义, 75

