SUSE Linux Enterprise Server 15 SP4

储存管理指南

储存管理指南

SUSE Linux Enterprise Server 15 SP4

本指南提供关于如何管理 SUSE Linux Enterprise Server 上的储存设备的信息。

出版日期： 2025 年 4 月 3 日

https://documentation.suse.com

版权所有 © 2006–2025 SUSE LLC 和贡献者。保留所有权利。

根据 GNU 自由文档许可 (GNU Free Documentation License) 版本 1.2 或（根据您的选择）版本 1.3 中的条款，在此授予您复制、分发和/或修改本文档的权限；本版权声明和许可附带不可变部分。许可版本 1.2 的副本包含在题为“GNU Free Documentation License”的部分。

有关 SUSE 商标，请参见 https://www.suse.com/company/legal/。所有其他第三方商标分别为相应所有者的财产。商标符号（®、™ 等）代表 SUSE 及其关联公司的商标。星号 (*) 代表第三方商标。

本指南力求涵盖所有细节，但这不能确保本指南准确无误。SUSE LLC 及其关联公司、作者和译者对于可能出现的错误或由此造成的后果皆不承担责任。

目录

前言 xiv

1 可用文档 xiv

2 改进文档 xiv

3 文档约定 xv

4 支持 xvii

SUSE Linux Enterprise Server 支持声明 xvii · 技术预览 xviii

I 文件系统和挂载 1

1 Linux 中文件系统的概述 2

1.1 术语 3

1.2 Btrfs 3

主要功能： 3 · SUSE Linux Enterprise Server 上的根文件系统设置 4 · 从 ReiserFS 和 ext 文件系统迁移到 Btrfs 8 · Btrfs 管理 9 · Btrfs 子卷定额支持 10 · Btrfs 上的交换 13 · Btrfs 发送/接收 13 · 数据去重支持 17 · 从根文件系统删除子卷 17

1.3 XFS 19

XFS 格式 20

1.4 Ext2 20

1.5 Ext3 21

可方便并高度可靠地从 ext2 升级 21 · 将 ext2 文件系统转换为 ext3 22

1.6 Ext4 22

可靠性和性能 23 · Ext4 文件系统 inode 大小及 inode 数量 23 · 升级到 Ext4 26

1.7 ReiserFS 27

1.8 OpenZFS 和 ZFS 27

1.9 其他受支持的文件系统 27

1.10 已阻止的文件系统 28

1.11 Linux 中的大型文件支持 29

1.12 Linux 内核储存的限制 31

1.13 释放未使用的文件系统块 31
定期 TRIM 32 • 联机 TRIM 33

1.14 文件系统查错 33
Btrfs 错误：设备上没有剩余空间 33 • Btrfs：跨设备平衡数据 35 • 不要在 SSD 中进行碎片整理 36

1.15 更多信息 36

2 调整文件系统的大小 38

2.1 使用案例 38

2.2 调整大小指导原则 38
支持调整大小的文件系统 39 • 增加文件系统的大小 39 • 减小文件系统的大小 39

2.3 更改 Btrfs 文件系统的大小 40

2.4 更改 XFS 文件系统的大小 41

2.5 更改 ext2、ext3 或 ext4 文件系统的大小 41

3 挂载储存设备 43

3.1 了解 UUID 43

3.2 udev 的永久设备名称 43

3.3 挂载网络储存设备 44

4 用于块设备操作的多层缓存 45

4.1 一般术语 45

4.2 缓存模式 46

4.3 bcache 47

主要功能 47 · 设置 bcache 设备 47 · 使用 sysfs 配置 bcache 49

4.4 lvmcache 49

配置 lvmcache 49 · 去除缓存池 51

II 逻辑卷 (LVM) 53

5 LVM 配置 54

5.1 了解逻辑卷管理器 54

5.2 创建卷组 56

5.3 创建逻辑卷 59

精简配置的逻辑卷 62 · 创建镜像卷 63

5.4 自动激活非根 LVM 卷组 64

5.5 调整现有卷组的大小 65

5.6 调整逻辑卷的大小 66

5.7 删除卷组或逻辑卷 68

5.8 使用 LVM 命令 68

使用命令调整逻辑卷的大小 72 · 使用 LVM 缓存卷 73

5.9 标记 LVM2 储存对象 74

使用 LVM2 标记 74 · 创建 LVM2 标记的要求 75 · 命令行标记语法 75 · 配置文件语法 76 · 将标记用于群集中的简单激活控制 78 · 使用标记激活群集中的首选主机 78

6 LVM 卷快照 82

6.1 了解卷快照 82

6.2 使用 LVM 创建 Linux 快照 83

6.3 监视快照 83

6.4 删除 Linux 快照 84

6.5 在虚拟主机上使用虚拟机的快照 85

6.6 将快照与来源逻辑卷合并以还原更改或回滚到先前的状态 86

III 软件 RAID 89

7 软件 RAID 配置 90

7.1 了解 RAID 级别 90

RAID 0 90 • RAID 1 91 • RAID 2 和 RAID 3 91 • RAID 4 91 • RAID 5 91 • RAID 6 92 • 嵌套和复杂 RAID 级别 92

7.2 使用 YaST 配置软件 RAID 92

RAID 名称 94

7.3 在 AArch64 上的 RAID 5 中配置条带大小 95

7.4 监视软件 RAID 96

7.5 更多信息 96

8 为根分区配置软件 RAID 97

8.1 针对根分区使用软件 RAID 设备的先决条件 97

8.2 为根 (/) 分区设置具有软件 RAID 设备的系统 97

9 创建软件 RAID 10 设备 104

9.1 使用 **mdadm** 创建嵌套 RAID 10 设备 104

使用 mdadm 创建嵌套的 RAID 10 (1+0) 105 • 使用 mdadm 创建嵌套的 RAID 10 (0+1) 107

9.2 创建复杂 RAID 10 109

复杂 RAID 10 中的设备和复本数 110 • 布局 110 • 使用 YaST 分区程序创建复杂 RAID 10 113 • 使用 mdadm 创建复杂 RAID 10 116

10 创建降级 RAID 阵列 118

11 使用 mdadm 调整软件 RAID 阵列的大小 120

11.1 增加软件 RAID 的大小 121

增加组件分区的大小 121 • 增加 RAID 阵列的大小 122 • 增加文件系统的大小 124

11.2 减小软件 RAID 的大小 124

减小文件系统的大小 124 • 减小 RAID 阵列的大小 125 • 减小组件分区的大小 126

12 适用于 MD 软件 RAID 的储存机箱 LED 实用程序 128

12.1 储存机箱 LED 监视服务 128

12.2 储存机箱 LED 控制应用程序 130

模式名称 131 • 设备列表 134 • 示例 135

12.3 更多信息 135

13 软件 RAID 查错 136

13.1 修复故障磁盘之后进行恢复 136

IV 网络储存 138

14 iSNS for Linux 139

14.1 iSNS 的工作原理 139

14.2 安装 iSNS Server for Linux 141

14.3 配置 iSNS 发现域 142

创建 iSNS 发现域 142 • 向发现域添加 iSCSI 节点 144

14.4 启动 iSNS 服务 145

14.5 更多信息 145

15 经由 IP 网络的大容量储存：iSCSI 146

15.1 安装 iSCSI LIO 目标服务器和 iSCSI 发起端 147

15.2 设置 iSCSI LIO 目标服务器 147

iSCSI LIO 目标服务启动和防火墙设置 148 • 配置身份验证以发现 iSCSI LIO 目标和发起端 149 • 准备储存空间 150 • 设置 iSCSI LIO 目标组 151 • 修改 iSCSI LIO 目标组 155 • 删除 iSCSI LIO 目标组 155

15.3 配置 iSCSI 发起端 156

使用 YaST 配置 iSCSI 发起端 156 • 手动设置 iSCSI 发起端 159 • iSCSI 发起端数据库 160

15.4 使用 targetcli-fb 设置软件目标 162

15.5 安装时使用 iSCSI 磁盘 167

15.6 iSCSI 查错 167

在 iSCSI LIO 目标服务器上设置目标 LUN 时发生门户错误 167 • iSCSI LIO 目标在其他计算机上不可见 168 • iSCSI 通讯的数据包被丢弃 168 • 将 iSCSI 卷与 LVM 配合使用 168 • 配置文件设置为手动时，会挂载 iSCSI 目标 169

15.7 iSCSI LIO 目标术语 169

15.8 更多信息 171

16 以太网光纤通道储存：FCoE 172

16.1 在安装过程中配置 FCoE 接口 173

16.2 安装 FCoE 和 YaST FCoE 客户端 174

16.3 使用 YaST 管理 FCoE 服务 174

16.4 使用命令配置 FCoE 177

16.5 使用 FCoE 管理工具管理 FCoE 实例 178

16.6 更多信息 180

17 NVMe over Fabric 182

17.1 概览 182

17.2 设置 NVMe over Fabric 主机 182

安装命令行客户端 182 · 发现 NVMe over Fabric 目标 183 · 连接 NVMe over Fabric 目标 183 · 多路径 184

17.3 设置 NVMe over Fabric 目标 185

安装命令行客户端 185 · 配置步骤 185 · 备份和恢复目标配置 188

17.4 特殊硬件配置 189

概览 189 · Broadcom 189 · Marvell 189

17.5 更多信息 190

18 管理设备的多路径 I/O 191

18.1 了解多路径 I/O 191

多路径术语 191

18.2 硬件支持 192

多路径实现：设备映射程序和 NVMe 192 · 针对多路径的储存阵列自动检测 193 · 需要特定硬件处理程序的储存阵列 193

18.3 规划多路径 193
先决条件 194 · 多路径安装类型和 initramfs 194 · 磁盘管理任务 196 · 软件 RAID 和复杂的储存堆栈 196 · 高可用性解决方案 196

18.4 多路径管理工具 197
设备映射程序多路径模块 197 · 多路径 I/O 管理工具 198 · 多路径设备上的 MD RAID 198 · **multipathd** 守护程序和 **multipath** 命令 199 · multipath 命令 201 · mpathpersist 实用程序 202

18.5 针对多路径配置系统 204
启用、启动和停止多路径服务 204 · 针对多路径准备 SAN 设备 205 · 多路径设备上的分区 206

18.6 多路径配置 206
创建 /etc/multipath.conf 文件 207 · multipath.conf 语法 207 · `/etc/multipath.conf` 部分 208 · 应用 `/etc/multipath.conf` 修改 209 · 生成 WWID 210

18.7 配置巡回检测、排队及故障回复的策略 210

18.8 将非多路径设备加入黑名单 212

18.9 配置用户用好的名称或别名 216
HA 群集中的多路径设备名称 220

18.10 配置路径故障转移策略和优先级 221
配置路径故障转移策略 222 · 配置故障转移优先级 223 · 报告目标路径组 230

18.11 配置根设备的多路径 I/O 231
安装时启用多路径 I/O 231 · 为现有根设备启用多路径 I/O 233 · 在根设备上禁用多路径 I/O 234

18.12 为现有软件 RAID 配置多路径 I/O 234

18.13 在多路径设备上使用 LVM2 237

18.14 最佳实践 237
在不重引导的情况下扫描新设备 237 • 在不重引导的情况下扫描新的分区设备 239 • 查看多路径 I/O 状态 241 • 在错误状况下管理 I/O 242 • 解决停止的 I/O 244 • 配置 IBM Z 设备的默认设置 244 • 将多路径与 NetApp 设备配合使用 245 • 将 --noflush 与多路径设备配合使用 245 • 根设备为多路径时的 SAN 超时设置 246

18.15 MPIO 查错 246
在多路径设备上安装 GRUB2 246 • 启用多路径时系统在引导过程中退出到紧急外壳 246 • 升级到多路径 0.4.9 或更高版本后单个设备的 PRIO 设置不起作用 250 • 升级到 multipath-tools-0.4.9 或更高版本后带有参数的 PRIO 设置不起作用 250 • 技术信息文档 251

19 通过 NFS 共享文件系统 252

19.1 概览 252

19.2 安装 NFS 服务器 253

19.3 配置 NFS 服务器 254
使用 YaST 导出文件系统 254 • 手动导出文件系统 255 • 采用 Kerberos 的 NFS 258

19.4 配置客户端 258
使用 YaST 导入文件系统 259 • 手动导入文件系统 260 • 并行 NFS (pNFS) 261

19.5 管理 NFSv4 访问控制列表 263

19.6 更多信息 264

19.7 收集信息以供 NFS 查错 264
常见查错 264 • 高级 NFS 调试 266

20 Samba 269

20.1 术语 269

20.2 安装 Samba 服务器 271

20.3 启动和停止 Samba 271

20.4 配置 Samba 服务器 271

使用 YaST 配置 Samba 服务器 271 • 手动配置服务器 274

20.5 配置客户端 278

使用 YaST 配置 Samba 客户端 278 • 在客户端上挂载 SMB1/CIFS 共享 278

20.6 将 Samba 用作登录服务器 280

20.7 配置了 Active Directory 的网络中的 Samba 服务器 281

20.8 高级主题 282

使用 systemd 自动挂载 CIFS 文件系统 282 • Btrfs 上的透明文件压缩 283 • 快照 285

20.9 更多信息 293

21 使用 Autofs 按需挂载 295

21.1 安装 295

21.2 配置 295

Master 映射文件 295 • 映射文件 297

21.3 操作和调试 298

控制 autofs 服务 298 • 调试自动挂载器问题 299

21.4 自动挂载 NFS 共享 300

21.5 高级主题 301

/net 挂载点 301 • 使用通配符自动挂载子目录 301 • 自动挂载 CIFS 文件系统 302

A GNU 许可证 303

前言

1 可用文档

联机文档

https://documentation.suse.com/#sles 上提供了此产品的相关联机文档。您可浏览或下载各种格式的文档。

https://documentation.suse.com/ 上提供了其他产品的相关联机文档。

注意：最新更新

最新的文档更新通常会在文档的英文版中提供。

发行说明

有关发行说明，请参见 https://www.suse.com/releasenotes/。

在您的系统中

要以脱机方式使用，请参见安装的系统中 `/usr/share/doc` 下的文档。许多命令的**手册页**中也对相应命令进行了详细说明。要查看手册页，请运行 **`man`** 后跟特定的命令名。如果系统上未安装 **`man`** 命令，请使用 **`sudo zypper install man`** 加以安装。

2 改进文档

欢迎您提供针对本文档的反馈及改进建议。您可以通过以下渠道提供反馈：

服务请求和支持

有关产品可用的服务和支持选项，请参见 https://www.suse.com/support/。

要创建服务请求，需在 SUSE Customer Center 中注册订阅的 SUSE 产品。请转到 https://scc.suse.com/support/requests 并登录，然后点击新建。

Bug 报告

在 https://bugzilla.suse.com/ 中报告文档问题。要简化此过程，可以使用本文档 HTML 版本中的报告问题链接。将光标置于受影响的句子中，然后在右侧导航面板的提供反馈部分单击报告问题。这样会在 Bugzilla 中预先选择正确的产品和类别，并添加当前章节的链接。然后，您便可以立即开始键入 Bug 报告。需要一个 Bugzilla 帐户。

贡献

要帮助改进本文档，请使用本文档 HTML 版本中的 Edit source document（编辑源文档）链接。将光标置于受影响的句子中，然后在右侧导航面板的提供反馈部分单击 Edit source document（编辑源文档）。这样您将会转到 GitHub 上的源代码，可以在其中提出拉取请求。需要一个 GitHub 帐户。

> **注意：Edit source document（编辑源文档）仅适用于英语版本**
>
> Edit source document（编辑源文档）链接仅适用于每个文档的英语版本。对于其他所有语言，请按上文所述使用报告问题链接。

有关本文档使用的文档环境的详细信息，请参见软件源的自述文件（网址：https://github.com/SUSE/doc-sle/blob/main/README.adoc）

邮件

您也可以将有关本文档中的错误以及相关反馈发送至：`doc-team@suse.com`。请在其中包含文档标题、产品版本和文档发布日期。此外，请包含相关的章节号和标题（或者提供 URL），并提供问题的简要说明。

3 文档约定

本文档中使用了以下通知和排版约定：

- `/etc/passwd`：目录名称和文件名
- PLACEHOLDER：PLACEHOLDER 将会替换为实际的值
- PATH：环境变量 PATH

- **ls**、--help：命令、选项和参数
- user：用户或组
- package name：软件包的名称
- Alt 、 Alt – F1 ：按键或组合键；这些键以大写形式显示，如在键盘上一样
- 文件、 文件 › 另存为：菜单项，按钮
- AMD/Intel 本段内容仅与 AMD64/Intel 64 体系结构相关。箭头标记文本块的开始位置和结束位置。
  IBM Z, POWER 本段内容仅与 IBM Z 和 POWER 体系结构相关。箭头标记文本块的开始位置和结束位置。
- **跳舞的企鹅**（**企鹅**一章，↑其他手册）：此内容参见自其他手册中的一章。
- 必须使用 root 特权运行的命令。您往往还可以在这些命令前加上 **sudo** 命令，以非特权用户身份来运行它们。

```
# command
> sudo command
```

- 可以由非特权用户运行的命令。

```
> command
```

- 注意

警告：警报通知

在继续操作之前，您必须了解的不可或缺的信息。向您指出有关安全问题、潜在数据丢失、硬件损害或物理危害的警告。

重要：重要通知

在继续操作之前，您必须了解的重要信息。

注意：注意通知

额外信息，例如有关软件版本差异的信息。

提示：提示通知

有用信息，例如指导方针或实用性建议。

4 支持

下面提供了 SUSE Linux Enterprise Server 的支持声明和有关技术预览的一般信息。有关产品生命周期的细节，请参见《升级指南》, 第 2 章 “生命周期和支持”。

如果您有权获享支持，可在《管理指南》, 第 47 章 “收集系统信息以供支持所用”中查找有关如何收集支持票据所需信息的细节。

4.1 SUSE Linux Enterprise Server 支持声明

要获得支持，您需要一个适当的 SUSE 订阅。要查看为您提供的具体支持服务，请转到 https://www.suse.com/support/ 并选择您的产品。

支持级别的定义如下：

L1

问题判定，该技术支持级别旨在提供兼容性信息、使用支持、持续维护、信息收集，以及使用可用文档进行基本查错。

L2

问题隔离，该技术支持级别旨在分析数据、重现客户问题、隔离问题区域，并针对级别 1 不能解决的问题提供解决方法，或作为级别 3 的准备级别。

L3

问题解决，该技术支持级别旨在借助工程方法解决级别 2 支持所确定的产品缺陷。

对于签约的客户与合作伙伴，SUSE Linux Enterprise Server 将为除以下软件包外的其他所有软件包提供 L3 支持：

- 技术预览。
- 声音、图形、字体和作品。
- 需要额外客户合同的软件包。
- **Workstation Extension** 模块随附的某些软件包仅享受 L2 支持。
- 名称以 `-devel` 结尾的软件包（包含头文件和类似的开发人员资源）只能与其主软件包一起获得支持。

SUSE 仅支持使用原始软件包，即，未发生更改且未重新编译的软件包。

4.2 技术预览

技术预览是 SUSE 提供的旨在让用户大致体验未来创新的各种软件包、堆栈或功能。随附这些预览只是为了提供方便，让您有机会在自己的环境中测试新的技术。非常希望您能提供反馈！如果您测试了技术预览，请联系 SUSE 代表，将您的体验和用例告知他们。您的反馈对于我们的未来开发非常有帮助。

但是，技术预览存在以下限制：

- 技术预览仍处于开发阶段。因此，它们的功能可能不完备、不稳定，或者在其他方面**不**适合用于生产。
- 技术预览**不**受支持。
- 技术预览可能仅适用于特定的硬件体系结构。
- 技术预览的细节和功能可能随时会发生变化。因此，可能无法升级到技术预览的后续版本，而只能进行全新安装。
- 我们随时可能会放弃技术预览。例如，如果 SUSE 发现某个预览不符合客户或市场的需求，或者不能证明它符合企业标准，则可能会放弃该预览。SUSE 不承诺未来将提供此类技术的受支持版本。

有关产品随附的技术预览的概述，请参见 https://www.suse.com/releasenotes/ 上的发行说明。

I 文件系统和挂载

1 Linux 中文件系统的概述 2

2 调整文件系统的大小 38

3 挂载储存设备 43

4 用于块设备操作的多层缓存 45

1 Linux 中文件系统的概述

SUSE Linux Enterprise Server 随附了不同的文件系统供您选择，包括 Btrfs、Ext4、Ext3、Ext2 和 XFS。每个文件系统都有各自的优点和缺点。有关 SUSE Linux Enterprise Server 中主要文件系统的并排功能比较，请参见 https://www.suse.com/releasenotes/x86_64/SUSE-SLES/15-SP3/#file-system-comparison（支持的文件系统比较）。本章概述了这些文件系统的工作原理以及它们的优点。

在 SUSE Linux Enterprise 12 中，Btrfs 是操作系统的默认文件系统，XFS 是所有其他使用案例的默认文件系统。此外，SUSE 仍继续支持 Ext 系列的文件系统以及 OCFS2。根据默认设置，Btrfs 文件系统将设置为使用子卷。对于使用 snapper 基础结构的根文件系统，将会自动启用快照。有关 snapper 的详细信息，请参见《管理指南》, 第 10 章 “通过 Snapper 进行系统恢复和快照管理”。

专业的高性能设置可能需要高可用性的储存系统。为符合高性能群集案例的要求，SUSE Linux Enterprise Server 在 High Availability Extension 附加产品中加入了 OCFS2（Oracle Cluster File System 2) 与 Distributed Replicated Block Device (DRBD)。本指南中不涉及这些高级储存系统的内容。有关信息，请参见 《SUSE Linux Enterprise High Availability Extension 管理指南》 (https://documentation.suse.com/sle-ha-15/html/SLE-HA-all/book-administration.html)。

记住一点很重要：没有任何一种文件系统适合所有种类的应用。每个文件系统都有各自的特定优点和缺点，必须将这些因素考虑在内。此外，即使是最复杂的文件系统也不能替代合理的备份策略。

本节中使用的术语**数据完整性**和**数据一致性**并不是指用户空间数据（您的应用程序写入其文件的数据）的一致性。此数据是否一致必须由应用程序本身控制。

除非本节特别指明，否则设置或更改分区以及文件系统所需的一切步骤，都可以使用 YaST 分区程序（强烈推荐使用）来执行。有关信息，请参见《部署指南》, 第 10 章 “专家分区程序”。

1.1 术语

元数据

文件系统内部的一种数据结构。它可确保磁盘上的所有数据都有条不紊，并且可供访问。几乎每一种文件系统都有自己的元数据结构，这也是文件系统展现出不同性能特性的原因所在。维护元数据的完整性非常重要，因为如果不这样，则可能无法访问文件系统中的所有数据。

inode

文件系统的数据结构包含文件的各种信息，包括大小、链接数量、实际储存文件内容的磁盘块的指针、创建、修改和访问的日期与时间。

日志

在提及文件系统时，日志是包含某种日志的磁盘上结构，文件系统将要对文件系统的元数据进行的更改储存在此日志中。日志可大大降低文件系统的恢复时间，因为有了它就不需要在系统启动时执行文件系统全面检查这一冗长的搜索程序。而只是重放日志。

1.2 Btrfs

Brtfs 是由 Chris Mason 开发的一种写时复制 (COW) 文件系统。它基于 Ohad Rodeh 开发的适用于 COW 的 B 树。Btrfs 是日志记录样式的文件系统。它不记录块更改，而是将块更改写入新位置，然后链接上更改。新更改在上一次写后才提交。

1.2.1 主要功能：

Btrfs 提供容错、修复和易于管理的功能，比如：

- 可写快照，允许应用更新后按需轻松回滚系统或允许备份文件。
- 子卷支持：Btrfs 会在为其指派的空间池中创建默认子卷。它允许您在相同空间池中创建更多的子卷，作为不同的文件系统。子卷的数目仅受分配给池的空间所限。
- Btrfs 命令行工具中提供了在线检查和修复功能 **`scrub`**。它会在假设树状结构没有问题的前提下，验证数据和元数据的完整性。您可以在安装的文件系统上定期运行 scrub；正常操作期间，它将在后台运行。

- 不同 RAID 级别，适用于元数据和用户数据。
- 用于元数据和用户数据的不同校验和，可改进错误检测。
- 与 Linux 逻辑卷管理器 (LVM) 储存对象集成。
- 与 SUSE Linux Enterprise Server 上的 YaST 分区程序及 AutoYaST 整合。这还包括在多个设备 (MD) 和设备映射程序 (DM) 储存配置上创建 Btrfs 文件系统。
- 从现有的 Ext2、Ext3 以及 Ext4 文件系统进行脱机迁移。
- `/boot` 的引导加载程序支持，允许从 Btrfs 分区引导。
- SUSE Linux Enterprise Server }15 SP4 中的 RAID0、RAID1 和 RAID10 配置文件支持多卷 Btrfs。更高的 RAID 级别尚不受支持，但安装将来发布的服务包后可能会支持。
- 使用 Btrfs 命令设置透明压缩。

1.2.2 SUSE Linux Enterprise Server 上的根文件系统设置

SUSE Linux Enterprise Server 默认设置为对根分区使用 Btrfs 和快照。快照可让您在应用更新之后有需要时轻松地回滚系统，也可让您备份文件。快照可通过 SUSE Snapper 基础结构轻松管理，如《管理指南》, 第 10 章 “通过 Snapper 进行系统恢复和快照管理”所述。有关 SUSE Snapper 项目的一般信息，请参见 OpenSUSE.org (http://snapper.io) 上的 Snapper 门户网站 Wiki。

使用快照回滚系统时，必须确保在回滚期间，数据（例如用户的主目录、Web 和 FTP 服务器内容或日志文件）不会遗失或被重写。这一点通过使用根文件系统上的 Btrfs 子卷实现。子卷可从快照中排除。安装期间，根据 YaST 建议，SUSE Linux Enterprise Server 上的默认根文件系统设置包含下列子卷。由于以下原因，它们会从快照中排除。

`/boot/grub2/i386-pc`、`/boot/grub2/x86_64-efi`、`/boot/grub2/powerpc-ieee1275`、`/boot/grub2/s390x-emu`

不能回滚引导加载程序配置。上面列出的目录是架构专属目录。前两个目录位于 AMD64/Intel 64 计算机上，后两个目录分别位于 IBM POWER 和 IBM Z 上。

`/home`

如果独立的分区中没有 `/home`，便会将该目录排除以免在回滚时发生数据丢失。

`/opt`

第三方产品通常安装到 `/opt` 下。排除此目录是为了防止在回滚时卸装这些应用程序。

`/srv`

包含 Web 和 FTP 服务器的数据。排除此目录是为了防止在回滚时发生数据丢失。

`/tmp`

包含临时文件和缓存的所有目录都会排除在快照范围之外。

`/usr/local`

在手动安装软件时会用到此目录。系统会将该目录排除以免在回滚时卸载这些安装的软件。

`/var`

此目录包含许多变量文件（包括日志、暂时缓存、`/var/opt` 中的第三方产品），是虚拟机映像和数据库的默认位置。因此，创建此子卷是为了从快照中排除所有这些变量数据，且已禁用“写入时复制”。

警告：回滚支持

仅当您未移除任何预先配置的子卷时，SUSE 才支持回滚。不过，您可以使用 YaST 分区程序添加子卷。

1.2.2.1 挂载压缩的 Btrfs 文件系统

Btrfs 文件系统支持透明压缩。如果启用，Btrfs 将在写入时压缩文件数据，并在读取时解压缩文件数据。

使用 `compress` 或 `compress-force` 挂载选项，并选择压缩算法 `zstd`、`lzo` 或 `zlib`（默认）。zlib 压缩的压缩率更高，而 lzo 的压缩速度更快，并且所需的 CPU 负载更小。zstd 算法提供了一种新式折衷方案，其性能接近 lzo，而压缩率与 zlib 类似。

例如：

```
# mount -o compress=zstd /dev/sdx /mnt
```

如果您创建了一个文件并在其中写入数据，而压缩后的结果大于或等于未压缩时的大小，则将来针对此文件执行写入操作后，Btrfs 会始终跳过压缩。如果您不希望有这种行为，请使用 `compress-force` 选项。对于包含一些初始不可压缩数据的文件而言，此选项可能很有用。

请注意，压缩只会作用于新文件。如果使用 `compress` 或 `compress-force` 选项挂载文件系统，则在未压缩的情况下写入的文件将不会压缩。此外，包含 `nodatacow` 属性的文件的内容永远不会压缩：

```
# chattr +C FILE
# mount -o nodatacow  /dev/sdx /mnt
```

加密与任何压缩操作均无关。在此分区中写入一些数据后，请打印细节：

```
# btrfs filesystem show /mnt
btrfs filesystem show /mnt
Label: 'Test-Btrfs'  uuid: 62f0c378-e93e-4aa1-9532-93c6b780749d
        Total devices 1 FS bytes used 3.22MiB
      devid    1 size 2.00GiB used 240.62MiB path /dev/sdb1
```

如果您希望此设置是永久性的，请在 `/etc/fstab` 配置文件中添加 `compress` 或 `compress-force` 选项。例如：

```
UUID=1a2b3c4d /home btrfs subvol=@/home,compress 0 0
```

1.2.2.2 挂载子卷

在 SUSE Linux Enterprise Server 上，从快照进行系统回滚的程序通过先从快照引导来执行。这样一来，您便可以在运行回滚之前，在运行的同时检查快照。只要挂载子卷，就可以实现从快照引导（通常没必要）。

除了第 1.2.2 节 “SUSE Linux Enterprise Server 上的根文件系统设置”中列出的子卷之外，系统中还存在一个名为 `@` 的卷。这是默认的子卷，将作为根分区 (`/`) 挂载。其他子卷将挂载到此卷中。

从快照引导时，使用的不是 `@` 子卷，而是快照。快照中包括的文件系统部分将以只读方式挂载为 `/`。其他子卷将以可写入方式挂载到快照中。此状态默认为临时状态，下次重引导时将还原先前的配置。若要使它成为永久状态，请执行 **`snapper rollback`** 命令。这将使目前引导的快照成为新的**默认**子卷，在重引导之后将会使用它。

1.2.2.3 检查可用空间

通常可通过运行 **df** 命令来检查文件系统的用量。在 Btrfs 文件系统上，**df** 的输出可能有误导性，因为除了原始数据分配的空间以外，Btrfs 文件系统还会为元数据分配并使用空间。

因此，即使看上去仍有大量的可用空间，Btrfs 文件系统也可能会报告空间不足。发生这种情况时，为元数据分配的全部空间都已用尽。使用以下命令可检查 Btrfs 文件系统上已用和可用的空间：

btrfs filesystem show

```
> sudo btrfs filesystem show /
Label: 'ROOT'  uuid: 52011c5e-5711-42d8-8c50-718a005ec4b3
        Total devices 1 FS bytes used 10.02GiB
        devid    1 size 20.02GiB used 13.78GiB path /dev/sda3
```

显示文件系统的总大小及其用量。如果最后一行中的这两个值匹配，则表示文件系统上的全部空间都已分配出去。

btrfs filesystem df

```
> sudo btrfs filesystem df /
Data, single: total=13.00GiB, used=9.61GiB
System, single: total=32.00MiB, used=16.00KiB
Metadata, single: total=768.00MiB, used=421.36MiB
GlobalReserve, single: total=144.00MiB, used=0.00B
```

显示文件系统的已分配 (`total`) 空间和已用空间值。如果元数据的 `total` 和 `used` 值基本上相等，则表示元数据的全部空间都已分配出去。

btrfs filesystem usage

```
> sudo btrfs filesystem usage /
Overall:
    Device size:                  20.02GiB
    Device allocated:             13.78GiB
    Device unallocated:            6.24GiB
    Device missing:                  0.00B
    Used:                         10.02GiB
```

```
    Free (estimated):              9.63GiB      (min: 9.63GiB)
    Data ratio:                       1.00
    Metadata ratio:                   1.00
    Global reserve:              144.00MiB      (used: 0.00B)

             Data     Metadata  System
Id Path      single   single    single   Unallocated
-- --------- -------- --------- -------- -----------
 1 /dev/sda3 13.00GiB 768.00MiB 32.00MiB     6.24GiB
-- --------- -------- --------- -------- -----------
   Total     13.00GiB 768.00MiB 32.00MiB     6.24GiB
   Used       9.61GiB 421.36MiB 16.00KiB
```

显示类似于前两个命令输出合并所得的数据。

有关详细信息，请参见 **`man 8 btrfs-filesystem`** 和 https://btrfs.wiki.kernel.org/index.php/FAQ。

1.2.3 从 ReiserFS 和 ext 文件系统迁移到 Btrfs

您可以使用 **`btrfs-convert`** 工具，将数据卷从现有 ReiserFS 或 Ext（Ext2、Ext3 或 Ext4）迁移到 Btrfs 文件系统。该过程允许您对未挂载（脱机）的文件系统进行就地转换，执行此操作可能需要使用包含 **`btrfs-convert`** 工具的可引导安装媒体。该工具会在原始文件系统的可用空间内构建 Btrfs 文件系统，并直接链接到其中包含的数据。设备上必须有足够用于创建元数据的可用空间，否则转换将失败。原始文件系统将保持不变，Btrfs 文件系统不会占用任何可用空间。需要的空间大小取决于文件系统的内容，可能会因其中包含的文件系统对象（例如文件、目录、扩展属性）数量而异。由于系统会直接参照数据，文件系统上的数据量不会影响转换所需的空间，但使用尾部封装且大小超过 2 KiB 的文件除外。

警告：不支持根文件系统转换

不支持将根文件系统转换为 Btrfs，且不建议这样做。由于需要根据您的特定设置定制各种步骤，因此无法自动完成此类转换 — 该过程需要经过复杂的配置才能提供正确的回滚，`/boot` 必须位于根文件系统上，并且系统必须包含特定的子卷，等等。请保留现有的文件系统，或者从头开始重新安装整个系统。

要将原始文件系统转换为 Btrfs 文件系统，请运行：

```
# btrfs-convert /path/to/device
```

重要：检查 /etc/fstab

转换后，需确保 /etc/fstab 中对原始文件系统的所有参照都已调整，现指示设备包含 Btrfs 文件系统。

转换后，Btrfs 文件系统的内容将会反映源文件系统的内容。源文件系统将一直保留到您去除了在 fs_root/reiserfs_saved/image 中创建的相关只读映像为止。该映像文件实际上是转换前 ReiserFS 文件系统的一个“快照”，修改 Btrfs 文件系统时不会修改该映像。要去除该映像文件，请去除 reiserfs_saved 子卷：

```
# btrfs subvolume delete fs_root/reiserfs_saved
```

要将文件系统还原到原始文件系统，请使用以下命令：

```
# btrfs-convert -r /path/to/device
```

警告：更改将丢失

您在文件系统装载为 Btrfs 文件系统时所做的任何更改都将丢失。切勿在此期间执行任何平衡操作，否则将无法正确恢复文件系统。

1.2.4 Btrfs 管理

Btrfs 与 YaST 分区程序和 AutoYaST 集成。您可以在安装期间使用它来为根文件系统建立解决方案。安装之后，您可以使用 YaST 分区程序来查看和管理 Btrfs 卷。

btrfsprogs 软件包中提供了 Btrfs 管理工具。有关使用 Btrfs 命令的信息，请参见 **man 8 btrfs**、**man 8 btrfsck** 和 **man 8 mkfs.btrfs** 命令。有关 Btrfs 功能的信息，请参见 Btrfs Wiki，网址为 http://btrfs.wiki.kernel.org。

1.2.5 Btrfs 子卷定额支持

Btrfs 根文件系统子卷（例如 `/var/log`、`/var/crash` 或 `/var/cache`）在正常运作期间可能会使用所有可用的磁盘空间，如此将导致系统故障。为避免出现此状况，SUSE Linux Enterprise Server 提供了 Btrfs 子卷定额支持。只要根据 YaST 建议设置根文件系统，便可以启用和设置子卷定额。

1.2.5.1 使用 YaST 设置 Btrfs 定额

要使用 YaST 为根文件系统的子卷设置定额，请执行以下步骤：

1. 启动 YaST 并选择系统 › 分区程序，然后单击是确认警告。
2. 在左侧窗格中，单击 Btrfs。
3. 在主窗口中，选择要为其启用子卷定额的设备，然后单击底部的编辑。
4. 在编辑 Btrfs 窗口中，激活启用子卷定额复选框，然后单击下一步进行确认。

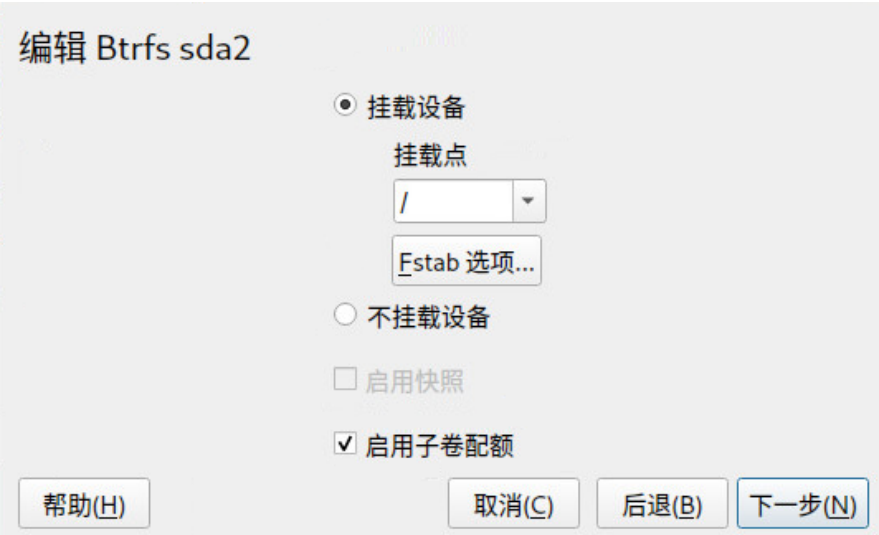


图 1.1： 启用 BTRFS 定额

5. 从现有子卷列表中，单击要按定额限制大小的子卷，然后单击底部的编辑。
6. 在编辑 Btrfs 的子卷窗口中，激活限制大小并指定最大引用大小。使用接受确认。

图 1.2： 设置子卷定额

新的大小限制将显示在子卷名称旁边：

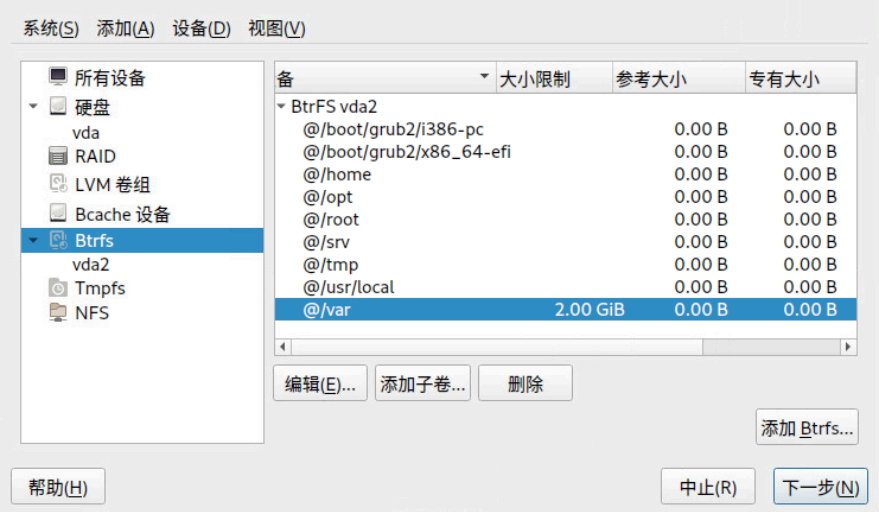


图 1.3： 设备的子卷列表

7. 单击下一步应用更改。

1.2.5.2 在命令行上设置 Btrfs 定额

要在命令行上设置根文件系统的子卷定额，请执行以下步骤：

1. 启用配额支持：

```
> sudo btrfs quota enable /
```

2. 取得子卷列表：

```
> sudo btrfs subvolume list /
```

只能为现有子卷设置配额。

3. 为上一步中所列的其中一个子卷设置配额。子卷可以用路径识别（例如 `/var/tmp`），也可以用 `0/子卷 ID` 识别（例如 `0/272`）。以下示例为 `/var/tmp` 设置 5 GB 配额。

```
> sudo btrfs qgroup limit 5G /var/tmp
```

大小单位可以是字节 (5000000000)、KB (5000000K)、MB (5000M) 或 GB (5G)。以字节为单位生成的值略有不同，因为 1024 字节 = 1 KiB，1024 KiB = 1 MiB，依此类推

4. 若要列出现有配额，请使用以下命令。`max_rfer` 列以字节为单位显示配额。

```
> sudo btrfs qgroup show -r /
```

提示：取消定额

如果您要取消现有定额，请将定额大小设置为 `none`：

```
> sudo btrfs qgroup limit none /var/tmp
```

若要禁用某个分区及其所有子卷的配额支持，请使用 **`btrfs quota disable`**：

```
> sudo btrfs quota disable /
```

1.2.5.3 更多信息

有关细节，请参见 **`man 8 btrfs-qgroup`** 和 **`man 8 btrfs-quota`**。Btrfs Wiki (https://btrfs.wiki.kernel.org/index.php/UseCases) 上的 UseCases 页面也提供了更多信息。

1.2.6 Btrfs 上的交换

重要：启用交换创建快照

如果源子卷包含任何已启用的交换文件，则您无法创建快照。

如果满足与生成的交换文件相关的以下准则，则 SLES 支持在 Btrfs 文件系统上的文件交换：

- 交换文件必须有 `NODATACOW` 和 `NODATASUM` 挂载选项。
- 不得压缩交换文件，您可以通过设置 `NODATACOW` 和 `NODATASUM` 挂载选项来确保满足此准则。两个选项都会禁用交换文件压缩。
- 在运行排它操作（例如设备调整大小、添加、去除或替换）时，或在运行平衡操作时，不能激活交换文件。
- 交换文件不能是稀疏文件。
- 交换文件不能是内联文件。
- 交换文件必须位于单个分配配置文件系统上。

1.2.7 Btrfs 发送/接收

Btrfs 允许生成快照来捕获文件系统的状态。例如，Snapper 可使用此功能在系统更改之前及之后创建快照，以便允许回滚。不过，将快照与发送/接收功能结合使用还可以在远程位置创建和维护文件系统的副本。例如，此功能可用于执行增量备份。

`btrfs send` 操作可计算同一子卷中两个只读快照之间的差异，并将差异发送到某个文件或 STDOUT。**`btrfs receive`** 操作会接收 send 命令的结果，并将其应用到快照。

1.2.7.1 先决条件

要使用发送/接收功能，需要满足以下要求：

- 源端（发送）和目标端（接收）各有一个 Btrfs 文件系统。
- Btrfs 发送/接收是对快照执行的，因此，相应的数据需要驻留在 Btrfs 子卷中。
- 源端中的快照必须为只读模式。
- SUSE Linux Enterprise 12 SP2 或更高版本。早期版本的 SUSE Linux Enterprise 不支持发送/接收。

1.2.7.2 增量备份

以下过程展示了 Btrfs 发送/接收操作的基本用法，其中示范了如何在 `/backup/data`（目标端）中创建 `/data`（源端）的增量备份。`/data` 必须是一个子卷。

过程 1.1： 初始设置

1. 在源端创建初始快照（在本示例中名为 `snapshot_0`），并确保将它写入该磁盘：

   ```
   > sudo btrfs subvolume snapshot -r /data /data/bkp_data
   sync
   ```

   即会创建一个新子卷 `/data/bkp_data`。该子卷将用作后续增量备份的基础，应将它保留为参照。

2. 将初始快照发送到目标端。由于这是初始的发送/接收操作，因此需要发送整个快照：

   ```
   > sudo bash -c 'btrfs send /data/bkp_data | btrfs receive /backup'
   ```

   目标端上即会创建一个新子卷 `/backup/bkp_data`。

完成初始设置后，可以创建增量备份，并将当前快照与先前快照之间的差异发送到目标端。操作过程始终是相同的：

1. 在源端创建新快照。
2. 将差异发送到目标端。
3. 可选：重命名和/或清理两端中的快照。

过程 1.2： **执行增量备份**

1. 在源端创建新快照，并确保将它写入该磁盘。在下面的示例中，快照命名为 bkp_data_当前日期：

```
> sudo btrfs subvolume snapshot -r /data /data/bkp_data_$(date +%F)
sync
```

创建新子卷，例如 `/data/bkp_data_2016-07-07`。

2. 将先前快照与您创建的快照之间的差异发送到目标端。为此，可以使用选项 `-p SNAPSHOT` 指定先前的快照。

```
> sudo bash -c 'btrfs send -p /data/bkp_data /data/bkp_data_2016-07-07 \
| btrfs receive /backup'
```

即会创建一个新子卷 `/backup/bkp_data_2016-07-07`。

3. 如此我们有了四个快照，每端各有两个：

`/data/bkp_data`
`/data/bkp_data_2016-07-07`
`/backup/bkp_data`
`/backup/bkp_data_2016-07-07`
现在，关于如何继续，您有三种选择：

- 保留两端中的所有快照。如果采用这种选择，您可以回滚到两端中的任一快照，同时会复制所有数据。不需要执行额外操作。执行后续增量备份时，请记得使用倒数第二个快照作为发送操作的父项。
- 仅保留源端中的最后一个快照，保留目标端中的所有快照。此外，允许回滚到两端中的任一快照 - 要回滚到源端中的特定快照，请对整个快照执行从目标端到源端的发送/接收操作。在源端执行删除/移动操作。
- 仅保留两端中的最后一个快照。采用这种方法，您会在目标端创建一个备份，该备份代表源端中生成的最后一个快照的状态。系统无法回滚到其他快照。在源端和目标端执行删除/移动操作。

a. 如果只想保留源端中的最后一个快照，请执行以下命令：

```
> sudo btrfs subvolume delete /data/bkp_data
> sudo mv /data/bkp_data_2016-07-07 /data/bkp_data
```

第一条命令将删除先前的快照，第二条命令将当前快照重命名为 `/data/bkp_data`。这可确保备份的最后一个快照始终命名为 `/data/bkp_data`。因此，您也可以始终使用此子卷名称作为增量发送操作的父项。

b. 如果只想保留目标端中的最后一个快照，请执行以下命令：

```
> sudo btrfs subvolume delete /backup/bkp_data
> sudo mv /backup/bkp_data_2016-07-07 /backup/bkp_data
```

第一条命令将删除先前的备份快照，第二条命令将当前备份快照重命名为 `/backup/bkp_data`。这可确保最新的备份快照始终命名为 `/backup/bkp_data`。

提示：发送到远程目标端

要将快照发送到远程计算机，请使用 SSH：

```
> btrfs send /data/bkp_data | ssh root@jupiter.example.com 'btrfs
 receive /backup'
```

1.2.8 数据去重支持

Btrfs 支持重复数据删除功能，具体办法是以指向通用储存位置中的块单一副本的逻辑链接替换文件系统中完全相同的块。SUSE Linux Enterprise Server 提供 **duperemove** 工具来扫描文件系统中有没有完全相同的块。在 Btrfs 文件系统上使用时，也可以用来删除这些重复的块，从而节省文件系统上的空间。系统上默认不会安装 **duperemove**。要使此功能可用，请安装软件包 duperemove。

注意：删除重复的大型数据集

如果您要删除重复的大量文件，请使用 `--hashfile` 选项：

```
> sudo duperemove --hashfile HASH_FILE file1 file2 file3
```

`--hashfile` 选项会将所有指定文件的哈希储存到 `HASH_FILE`（而不是 RAM 中），防止耗尽 RAM。`HASH_FILE` 可重复使用 - 当完成生成基线哈希文件的初始运行后，即可删除对大型数据集的重复更改。

duperemove 可以针对一系列文件操作，也可以以递归方式扫描某个目录：

```
> sudo duperemove OPTIONS file1 file2 file3
> sudo duperemove -r OPTIONS directory
```

它有两种操作模式：只读和重复数据删除。以只读模式运行时（即不使用 `-d` 开关），它会扫描给定文件或目录中的重复块，并将其打印出来。此模式适用于所有文件系统。

以重复数据删除模式执行 **duperemove** 仅在 Btrfs 文件系统上受支持。扫描给定文件或目录之后，将会提交重复的块以进行重复数据删除。

有关详细信息，请参见 **man 8 duperemove**。

1.2.9 从根文件系统删除子卷

出于特定目的，您可能需要从根文件系统中删除某个默认的 Btrfs 子卷。目的之一是将某个子卷（例如 `@/home` 或 `@/srv`）转换成单独设备上的文件系统。以下过程演示如何删除 Btrfs 子卷：

1. 确定要删除的子卷（例如 `@/opt`）。请注意，根路径始终使用子卷 ID“5”。

```
> sudo btrfs subvolume list /
ID 256 gen 30 top level 5 path @
ID 258 gen 887 top level 256 path @/var
ID 259 gen 872 top level 256 path @/usr/local
ID 260 gen 886 top level 256 path @/tmp
ID 261 gen 60 top level 256 path @/srv
ID 262 gen 886 top level 256 path @/root
ID 263 gen 39 top level 256 path @/opt
[...]
```

2. 查找托管根分区的设备名称：

```
> sudo btrfs device usage /
/dev/sda1, ID: 1
  Device size:            23.00GiB
  Device slack:              0.00B
  Data,single:             7.01GiB
  Metadata,DUP:            1.00GiB
  System,DUP:             16.00MiB
  Unallocated:            14.98GiB
```

3. 在单独的挂载点（例如 `/mnt`）上挂载根文件系统（ID 为 5 的子卷）：

```
> sudo mount -o subvolid=5 /dev/sda1 /mnt
```

4. 从挂载的根文件系统中删除 `@/opt` 分区：

```
> sudo btrfs subvolume delete /mnt/@/opt
```

5. 卸载以前挂载的根文件系统：

```
> sudo umount /mnt
```

1.3 XFS

SGI 在 20 世纪 90 年代初开始开发 XFS，最初计划将 XFS 作为 IRIX OS 的文件系统。开发 XFS 的目的是创建一个高性能的 64 位日志文件系统来满足对计算能力的极高要求。XFS 适合操纵大型文件，在高端硬件上表现优异。XFS 是 SUSE Linux Enterprise Server 中数据分区的默认文件系统。

快速回顾 XFS 的关键功能可解释为什么此文件系统经证明在高端计算方面是其他日志文件系统的强大竞争对手。

高可伸缩性

XFS 使用分配组来提供高可伸缩性

在创建 XFS 文件系统时，文件系统底层的块设备被分成 8 个或 8 个以上相同大小的线性区域。这些区域称为**分配组**。每个分配组管理自己的 inode 和可用空间。实际上，可以将分配组看作文件系统中的文件系统。因为分配组相互独立，所以内核可同时对多个分配组进行寻址。此功能是 XFS 优异的可伸缩性关键之所在。独立分配组的概念自然适合多处理器系统的需要。

性能较高

XFS 通过有效管理磁盘空间来提供高性能

可用空间和 inode 是由分配组内的 B^+ 树处理的。使用 B^+ 树将大大增强 XFS 的性能和可伸缩性。XFS 使用**延迟分配**，它可以通过将进程分为两部分而处理分配。将挂起事务储存在 RAM 中并保留适当数量的空间。XFS 仍不决定应储存数据的准确位置（即不指出文件系统块）。此决定将被延迟到最后的时刻。某些生存期很短的临时数据可能永远不会被储存到磁盘上，这是因为在 XFS 决定保存它们的实际位置时，这些数据可能已经过时了。以这种方式，XFS 增强了写性能并减少了文件系统分段。因为延迟分配引起写事件的频率比其他文件系统引起写事件的频率要低，所以如果写操作期间发生系统崩溃，则数据丢失可能会更加严重。

进行预分配以避免文件系统碎片

在将数据写入文件系统前，XFS **保留**（预分配）文件所需的可用空间。这样会大大减少文件系统碎片的数目。因为文件的内容不会分散在整个文件系统中，所以性能得以提高。

1.3.1 XFS 格式

SUSE Linux Enterprise Server 支持 XFS 文件系统的“磁盘格式” (v5)。这种格式的主要优点包括，所有 XFS 元数据的自动检查总数、文件类型支持以及支持文件更多数量的访问控制列表。

请注意，低于 3.12 版的 SUSE Linux Enterprise 内核、低于 3.2.0 版的 `xfsprogs` 以及在 SUSE Linux Enterprise 12 之前发布的 GRUB 2 版本均不支持这种格式。

重要：V4 已弃用

XFS 正在弃用采用 V4 格式的文件系统。此文件系统格式是由以下命令创建的：

```
mkfs.xfs -m crc=0 DEVICE
```

该格式在 SLE 11 和更低版本中使用，目前它通过 **`dmesg`** 创建警告消息：

```
Deprecated V4 format (crc=0) will not be supported after September 2030
```

如果您在 **`dmesg`** 命令的输出中看到上述消息，建议将文件系统更新到 V5 格式：

1. 将数据备份到另一台设备。
2. 在该设备上创建文件系统。

   ```
   mkfs.xfs -m crc=1 DEVICE
   ```

3. 从已更新的设备上的备份恢复数据。

1.4 Ext2

Ext2 的原身可以追溯到 Linux 历史的早期。其前身是“扩展文件系统”，于 1992 年 4 月实施，集成在 Linux 0.96c 中。扩展文件系统经历了数次修改，后来才称为 Ext2，曾经是多年来最受欢迎的 Linux 文件系统。但随着日志文件系统的创建以及其恢复时间的缩短，Ext2 的重要性逐渐降低。

简要总结 Ext2 的优点有助于您了解为什么它以前是（在某些领域现在仍是）许多 Linux 用户最喜欢使用的 Linux 文件系统。

可靠性和速度

Ext2 是一个“老古董”，它经历了许多改进和频繁的测试。这可能是人们经常称之为坚如磐石的文件系统的原因。在系统中断后，如果无法彻底卸装文件系统，则 e2fsck 将开始分析文件系统数据。系统使元数据恢复一致的状态，并将挂起的文件或数据块写入指定的目录（名为 `lost+found`）。与日志文件系统相比，e2fsck 会分析整个文件系统，而不仅仅是元数据中最近修改的位。这种操作所花的时间要远远超过检查日志文件系统的日志数据所花的时间。根据文件系统的大小，此过程可能需要半小时或更长时间。因此，对于任何要求高可用性的服务器，不要选择 Ext2。但是，因为 Ext2 不维护日志且使用的内存也更少，所以其速度往往快于其他文件系统。

可方便地升级

因为 Ext3 以 Ext2 代码为基础并且共享 Ext2 的磁盘上格式和元数据格式，所以从 Ext2 升级到 Ext3 非常容易。

1.5 Ext3

Ext3 由 Stephen Tweedie 设计。与所有其他下一代文件系统不同，Ext3 并没有采用全新的设计原则。它是在 Ext2 的基础上设计的。这两个文件系统密切关联。可以方便地在 Ext2 文件系统上建立 Ext3 文件系统。Ext2 和 Ext3 最重要的区别是 Ext3 支持日志。总之，Ext3 有三个主要优点：

1.5.1 可方便并高度可靠地从 ext2 升级

Ext2 的代码为 Ext3 奠定了坚实的基础，使后者成为受到高度评价的下一代文件系统。在 Ext3 中，它的可靠性和稳定性与日志文件系统的优点完美地结合在一起。不像转换至其他日志文件系统（例如 XFS）那么费时（备份整个文件系统，然后从头开始重新创建），转换到 Ext3 只需几分钟时间。升级到 Ext3 还很安全，因为从头重新创建整个文件系统可能会出现问题。考虑到等待升级到日志文件系统的现有 Ext2 系统的数量，就很容易明白为什么 Ext3 对许多系统管理员来说如此重要。从 Ext3 降级到 Ext2 与升级一样简单。将 Ext3 文件系统完全卸载，然后重新挂载成 Ext2 文件系统即可。

1.5.2 将 ext2 文件系统转换为 ext3

要将 Ext2 文件系统转换为 Ext3：

1. 作为 `root` 用户运行 **`tune2fs -j`** 来创建 Ext3 日志。
   此命令将用默认参数创建 Ext3 日志。
   要指定日志的大小和所驻留的设备，请改为运行 **`tune2fs`** `-J`，同时使用所需的日志选项 `size=` 和 `device=`。**`tune2fs`** 手册页中有关于 **`tune2fs`** 程序的更多信息。

2. 作为 `root` 用户编辑文件 `/etc/fstab`，以将为相应分区指定的文件系统类型从 `ext2` 更改为 `ext3`，然后保存更改。
   这确保可以正确识别出 Ext3 文件系统。此更改将在下次重引导后生效。

3. 若引导已设置为 Ext3 分区的根文件系统，请在 `initrd` 中添加模块 `ext3` 和 `jbd`。操作步骤如下：

   a. 打开或创建 `/etc/dracut.conf.d/filesystem.conf` 并添加以下行（请注意前导空格）：

   ```
   force_drivers+=" ext3 jbd"
   ```

   b. 然后运行 **`dracut`** `-f` 命令。

4. 重新启动系统。

1.6 Ext4

2006 年，Ext4 做为 Ext3 的传承面市。它是扩展的文件系统版本中的最新文件系统。Ext4 最初旨在增大储存大小，它支持最大大小为 1 EiB 的卷、最大大小为 16 TiB 的文件和无限个子目录。Ext4 使用区域（而不是传统的直接和间接块指针）来映射文件内容。使用区域可以改善在磁盘中存储数据以及从中检索数据的功能。

Ext4 还引入了多项性能增强功能，例如延迟块分配和速度大幅加快的文件系统检查例程。Ext4 还支持日志校验和，并可提供以纳秒度量的时间戳，因而更加可靠。Ext4 完全反向兼容于 Ext2 和 Ext3，后两个文件系统都可以作为 Ext4 挂载。

注意：Ext4 上的 Ext3 功能

Ext4 内核模块中的 Ext4 驱动程序完全支持 Ext3 功能。

1.6.1 可靠性和性能

某些其他日志文件系统采用“仅元数据”的日志方法。这意味着元数据始终保持一致的状态，但无法自动保证文件系统数据本身一致。Ext4 的设计既可以照顾到元数据，又可以照顾到数据。“照顾”的程度可以自定义。以 `data=journal` 模式挂载 Ext4 可以提供最高安全性（数据完整性），但由于要将元数据和数据都记入日志，因此可能会降低系统速度。另一种方法是使用 `data=ordered` 模式，此模式可确保数据和元数据的完整性，但只对元数据使用日志。文件系统驱动程序收集与一次元数据更新对应的所有数据块。这些数据块在更新元数据之前被写入磁盘中。这样，在不牺牲性能的情况下，元数据和数据的一致性得以实现。可使用的第三个挂载选项是 `data=writeback`，它允许数据在其元数据已经提交至日志后再写入主要文件系统。在性能方面，此选项常被认为是最佳选项。但它在维护内部文件系统完整性的同时，允许以前的数据在系统崩溃并恢复后再次出现在文件中。Ext4 使用 `data=ordered` 选项作为默认值。

1.6.2 Ext4 文件系统 inode 大小及 inode 数量

inode 用于存储文件的相关信息及其在文件系统中的块位置。为了让 inode 有空间可以容纳扩展属性以及 ACL，默认的 inode 大小已增大至 256 字节。

当您创建新的 Ext4 文件系统时，系统将根据可创建的 Inode 总数预先分配 Inode 表格中的空间。每 inode 的字节数比率以及文件系统的大小决定了可以创建的 inode 数量。建立文件系统时，将根据每 inode 字节数的单位空间创建 inode：

```
number of inodes = total size of the file system divided by the number of bytes
 per inode
```

inode 的数量控制着文件系统中可容纳的文件数：一个文件对应一个 inode。

重要：无法更改现有 Ext4 文件系统的 inode 大小

inode 分配完毕后，将无法更改 inode 大小的设置或每 inode 的字节数比率。如果不使用其他设置重新创建文件系统，或不扩展文件系统，则无法添加 Inode。超过 inode 最大数量时，只有删除部分文件才能在文件系统上创建新文件。

新建 Ext4 文件系统时，您可以指定 inode 大小和每 inode 的字节数比率，以控制文件系统上可容纳 inode 空间占用量以及文件数量。如果未指定块大小、inode 大小以及每 inode 的字节数比率值，则将应用 `/etc/mked2fs.conf` 中的默认值。有关信息，请参见 `mke2fs.conf(5)` 手册页。

使用以下指标：

- **inode 大小：** 默认的 inode 大小为 256 字节。指定字节值，即介于 128 字节（含）到块大小（含）之间的 2 的乘方值，如 128、256、512，以此类推。只有当 Ext4 文件系统上不使用扩展属性或 ACL 时才可使用 128 字节。
- **每 inode 的字节数比率：** 默认的每 inode 的字节数比率为 16384 字节。有效的每 inode 的字节数比率值必须是大于等于 1024 字节的 2 的乘方值，如 1024、2048、4096、8192、16384、32768，以此类推。该值不应小于文件系统的块大小，因为块大小是用于存储数据的最小空间大块。Ext4 文件系统的默认块大小为 4 KiB。此外，请考虑需要储存的文件数量及大小。例如，如果您的文件系统上将会有许多小文件，则可以指定一个较小的每 inode 的字节数比率，这样会增加 inode 的数量。如果文件系统将储存超大型文件，您可以指定一个较大的每 Inode 的字节数比率，这样可减少可能的 Inode 数。
  通常情况下，最好要保证有足够多的 inode 可供使用。如果 inode 数量过少且文件很小，则可能当磁盘上的文件数量已达最大值时实际上磁盘却还很空。如果 inode 过多且文件很大，则可能虽然报告仍有可用空间，但却无法使用，这是因为您无法在为 inode 预留的空间中新建文件。

使用以下任何一种方法设置 inode 大小以及每 inode 的字节数比率：

- **修改所有新 Ext4 文件系统的默认设置：** 在文本编辑器中，修改 `/etc/mke2fs.conf` 文件的 `defaults` 部分，以为 `inode_size` 和 `inode_ratio` 设置所需的默认值。这些值将应用到所有新的 Ext4 文件系统。例如：

```
blocksize = 4096
inode_size = 128
inode_ratio = 8192
```

- **在命令行处：** 新建 Ext4 文件系统时，将 inode 大小 (`-I 128`) 以及每 inode 的字节数比率 (`-i 8192`) 传递给 **`mkfs.ext4(8)`** 命令或 **`mke2fs(8)`** 命令。例如，使用以下任一命令：

```
> sudo mkfs.ext4 -b 4096 -i 8092 -I 128 /dev/sda2
> sudo mke2fs -t ext4 -b 4096 -i 8192 -I 128 /dev/sda2
```

- **在使用 YaST 安装期间：** 在安装期间，新建 Ext4 文件系统时要传递 inode 大小和每 inode 的字节数比率值。在专家分区程序中选择分区，然后单击编辑。在格式化选项下，选择格式化设备Ext4，然后单击选项。在格式化选项对话框中，从块大小（字节）、每 inode 的字节数和 Inode 大小下拉框中选择所需的值。
  例如，在块大小 (B) 下拉框中选择 4096、在每 inode 的字节数下拉框中选择 8192、在 Inode 大小下拉框中选择 128，然后单击确定。

1.6.3 升级到 Ext4

重要：备份数据

在对文件系统执行任何更新之前，请备份文件系统上的所有数据。

过程 1.3： **升级到 EXT4**

1. 要从 Ext2 或 Ext3 进行升级，必须启用以下功能：

   EXT4 所需的功能

   extents
   硬盘上的邻接块，用于使文件彼此相连并防止出现碎片

   unint_bg
   迟缓 inode 表初始化

   dir_index
   针对大目录的哈希 b 树查找

   在 Ext2 上： `as_journal`
   在 Ext2 文件系统上启用日志。

   要启用这些功能，请运行：

   - 在 Ext3 上：

     ```
     # tune2fs -O extents,uninit_bg,dir_index DEVICE_NAME
     ```

   - 在 Ext2 上：

     ```
     # tune2fs -O extents,uninit_bg,dir_index,has_journal DEVICE_NAME
     ```

2. 以 `root` 身份编辑 `/etc/fstab` 文件：将 `ext3` 或 `ext2` 记录更改为 `ext4`。此更改将在下次重引导后生效。

3. 要引导 Ext4 分区上设置的文件系统，请在 `initramfs` 中添加模块 `ext4` 和 `jbd`。打开或创建 `/etc/dracut.conf.d/filesystem.conf` 并添加以下行：

```
force_drivers+=" ext4 jbd"
```

需要通过运行以下命令来重写现有的 dracut `initramfs`：

```
dracut -f
```

4. 重引导系统。

1.7 ReiserFS

SUSE Linux Enterprise Server 15 中完全去除了 ReiserFS 支持。要将现有分区迁移到 Btrfs，请参见第 1.2.3 节 “从 ReiserFS 和 ext 文件系统迁移到 Btrfs”。

1.8 OpenZFS 和 ZFS

OpenZFS 和 ZFS 文件系统都不受 SUSE 的支持。ZFS 是一个闭源文件系统，因此不可由 SUSE 使用。用于约束 OpenZFS 的 CDDL 许可证与 GPL 许可证不兼容。但是，Btrfs 相应地为 OpenZFS 提供了一个优异的替代方案。ZFS 完全受 SUSE 的支持。

1.9 其他受支持的文件系统

表 1.1 “Linux 中的文件系统类型” 对 linux 支持的其他一些文件系统进行了总结。支持这些文件系统主要是为了确保与不同类型的媒体或异操作系统实现兼容和数据交换。

表 1.1： LINUX 中的文件系统类型

文件系统类型	说明
`iso9660`	CD-ROM 上的标准文件系统。
`msdos`	`fat`（最初由 DOS 使用的文件系统）现在已被多种操作系统采用。
`nfs`	网络文件系统：在此文件系统中，可以将数据储存在网络中的任何计算机上，并可以通过网络授予访问权限。

文件系统类型	说明
ntfs	Windows NT 文件系统；只读。
exfat	为使用闪存（例如 USB 闪存盘和 SD 卡）而优化的文件系统。
smbfs	Windows 等产品使用服务器消息块来支持通过网络启用文件访问。
ufs	供 BSD、SunOS 和 NextStep 使用。只在只读方式下支持此文件系统。
umsdos	MS-DOS 上的 Unix：在标准 fat 文件系统之上应用，通过创建特殊文件获得 Unix 功能（权限、链接和长文件名）。
vfat	虚拟 FAT：fat 文件系统的扩展（支持长文件名）。

1.10 已阻止的文件系统

出于安全原因，已阻止某些文件系统自动挂载。这些文件系统通常不再受到维护，并且不太常用。但是，可以装载这些文件系统的内核模块，因为内核中 API 仍是兼容的。将用户可挂载的文件系统和可卸设备上自动挂载的文件系统结合使用可能会导致非特权用户触发内核模块自动装载，并可能导致可卸设备储存潜在恶意的数据。

要获取不允许自动挂载的文件系统列表，请运行以下命令：

```
> sudo rpm -ql suse-module-tools  | sed -nE 's/.*blacklist_fs-(.*)\.conf/\1/p'
```

如果您尝试使用 **mount** 命令挂载包含已阻止文件系统的设备，该命令将输出错误消息，例如：

```
mount: /mnt/mx: unknown filesystem type 'minix' (hint: possibly blacklisted, see
 mount(8)).
```

要允许挂载文件系统，需要从阻止列表中去除特定的文件系统。每个已阻止的文件系统具有自身的配置文件，例如，efs 的配置文件是 /lib/modules.d/60-blacklist_fs-efs.conf。但是，请不要编辑这些文件，因为每当更新软件包 suse-module-tools 时，就会重写这些文件。要允许自动挂载已阻止的文件系统，可使用以下选项：

- 创建指向 /dev/null 的符号链接，例如，为 **efs** 文件系统创建以下符号链接：

```
> sudo ln -s /dev/null /etc/modules.d/60-blacklist_fs-efs.conf
```

- 将配置文件复制到 /etc/modprobe.d：

```
> sudo cp /lib/modules.d/60-blacklist_fs-efs.conf /etc/modprobe.d/60-
blacklist_fs-efs.conf
```

在配置文件中注释掉以下语句：

```
# blacklist omfs
```

即使无法自动挂载某个文件系统，您也可以直接使用 **modprobe** 装载该文件系统的相应内核模块：

```
> sudo modprobe FILESYSTEM
```

例如，对于 cramfs 文件系统，输出如下所示：

```
unblacklist: loading cramfs file system module
unblacklist: Do you want to un-blacklist cramfs permanently (<y>es/<n>o/
n<e>ver)? y
unblacklist: cramfs un-blacklisted by creating /etc/modprobe.d/60-blacklist_fs-
cramfs.conf
```

如果您选择 **yes**，则 **modprobe** 命令会调用一个脚本用于创建从所提供文件系统的配置文件到 /dev/null 的符号链接。因此，会从阻止列表中去除该文件系统。

1.11 Linux 中的大型文件支持

最初，Linux 支持的最大文件大小为 2 GiB（2^{31} 字节）。除非文件系统支持大型文件，否则 32 位系统上的最大文件大小为 2 GiB。

目前，所有标准文件系统都具有 LFS（大型文件支持）功能，理论上可支持的最大文件大小为 2^{63} 字节。表 1.2 “文件和文件系统的最大大小（磁盘格式，4 KiB 块大小）”概述了 Linux 文件和文件系统的当前磁盘上格式的限制。表中的数字基于文件系统使用 4 KiB 块大小的假设得出，

这是通用的标准。使用不同的块大小，结果也就不同。使用较稀疏的块时，表 1.2 “文件和文件系统的最大大小（磁盘格式，4 KiB 块大小）”中的最大文件大小可能会大于文件系统的实际大小。

注意：二进制倍数

在此文档中：1024 字节 = 1 KiB；1024 KiB = 1 MiB；1024 MiB = 1 GiB；1024 GiB = 1 TiB；1024 TiB = 1 PiB；1024 PiB = 1 EiB（另请参见 NIST：二进制倍数的前缀 (http://physics.nist.gov/cuu/Units/binary.html)）。

表 1.2： 文件和文件系统的最大大小（磁盘格式，4 KIB 块大小）

文件系统（4 KiB 块大小）	最大文件系统大小	最大文件大小
Btrfs	16 EiB	16 EiB
Ext3	16 TiB	2 TiB
Ext4	1 EiB	16 TiB
OCFS2（高可用性扩展中可使用的群集感知文件系统）	16 TiB	1 EiB
XFS	16 EiB	8 EiB
NFSv2（客户端）	8 EiB	2 GiB
NFSv3/NFSv4（客户端）	8 EiB	8 EiB

重要：限制

表 1.2 “文件和文件系统的最大大小（磁盘格式，4 KiB 块大小）”介绍了有关磁盘上格式的限制。Linux 内核自身的大小限制同样适用于其处理的文件和文件系统大小。下面介绍了这些限制：

文件大小

在 32 位系统上，文件不能超过 2 TiB（2^{41} 字节）。

文件系统大小

文件系统最大可以为 2^{73} 个字节。但是，目前可用的硬件尚不会超出这一限制。

1.12 Linux 内核储存的限制

表 1.3 “储存限制” 总结了与 SUSE Linux Enterprise Server 相关联的储存的内核限制。

表 1.3： 储存限制

储存功能	限制
支持的 LUN 最大数量	每个目标 16384 个 LUN。
每一个单独 LUN 的最大路径数量	默认情况下没有限制。每个路径视作一个常规 LUN。 每个目标的 LUN 数量以及每个 HBA 的目标数量决定了实际的限制（光纤通道 HBA 为 16777215）。
HBA 的最大数量	不限.实际限制取决于系统的 PCI 槽的数量。
每个操作系统使用 device-mapper-multipath 的最大路径数量（总计）	大约为 1024。实际数量取决于每个多路径设备的设备号字符串长度。它是 multipath-tools 中的一个编译时间变量，如果此限制会导致问题，则可提高其值。
每一个块设备的最大大小	最多 8 EiB。

1.13 释放未使用的文件系统块

在固态硬盘 (SSD) 和精简配置的卷中，释放未被文件系统使用的块会很有用。在支持 `unmap` 和 `TRIM` 操作的所有文件系统上，SUSE Linux Enterprise Server 完全支持执行这种操作。

有两种常用的 TRIM 操作 — 联机 TRIM 和定期 TRIM。释放设备的最合适方法取决于您的用例。一般情况下，建议使用定期 TRIM，尤其是当设备具有足够的可用块时。如果设备经常接近容量耗尽状态，则最好使用联机 TRIM。

重要：设备对 TRIM 操作的支持

在尝试使用 TRIM 操作之前，请始终校验您的设备是否支持该操作。否则，您可能会丢失该设备上的数据。要校验是否支持 TRIM 操作，请运行以下命令：

```
> sudo lsblk --discard
```

该命令会输出有关所有可用块设备的信息。如果 DISC-GRAN 和 DISC-MAX 列的值不为零，则表示设备支持 TRIM 操作。

1.13.1 定期 TRIM

定期 TRIM 由 systemd 定期调用的 **fstrim** 命令进行处理。您也可以手动运行该命令。

要安排定期 TRIM，请如下所示启用 fstrim.timer：

```
> sudo systemctl enable fstrim.timer
```

systemd 在 /usr/lib/systemd/system 中创建一个单元文件。默认情况下，该服务每周运行一次，这种频率通常已足够。但是，您可以通过将 OnCalendar 选项配置为所需值来更改频率。

fstrim 的默认行为是丢弃文件系统中的所有块。您可以在调用该命令时使用选项来修改此行为。例如，可以传递 offset 选项来定义释放过程的起始位置。有关细节，请参见 **man fstrim**。

fstrim 命令可对储存在 /etc/fstab 文件中的支持 TRIM 操作的所有设备执行释放 — 为此，请在调用该命令时使用 -A 选项。

要禁用特定设备的释放，请如下所示将选项 X-fstrim.notrim 添加到 /etc/fstab 文件中：

```
UID=83df497d-bd6d-48a3-9275-37c0e3c8dc74  /  btrfs  defaults,X-fstrim.notrim
                  0  0
```

1.13.2 联机 TRIM

每次向设备写入数据时，都会对该设备执行联机 TRIM。

要启用设备的联机 TRIM，请如下所示将 `discard` 选项添加到 `/etc/fstab` 文件中：

```
UID=83df497d-bd6d-48a3-9275-37c0e3c8dc74  /  btrfs  defaults,discard
```

或者，在 Ext4 文件系统上，可以使用 **`tune2fs`** 命令在 `/etc/fstab` 中设置 `discard` 选项：

```
> sudo tune2fs -o discard DEVICE
```

如果设备是结合 `discard` 选项通过 **`mount`** 挂载的，则也要将 `discard` 选项添加到 `/etc/fstab`：

```
> sudo mount -o discard DEVICE
```

注意：联机 TRIM 的缺点

使用 `discard` 选项可能会缩短某些低质量 SSD 设备的使用寿命。联机 TRIM 还可能会影响设备的性能，例如，在删除大量数据的情况下。在这种情况下，可能会重新分配一个擦除块，并在短时间后，再次将同一擦除块标记为未使用。

1.14 文件系统查错

本节说明文件系统的一些已知问题和可能的解决方案。

1.14.1 Btrfs 错误：设备上没有剩余空间

使用 Btrfs 文件系统的根 (`/`) 分区停止接受数据。您收到错误“设备上没有剩余空间”。

请参见下列各部分，了解有关此问题的可能原因和预防措施的信息。

1.14.1.1 Snapper 快照使用的磁盘空间

如果 Snapper 是针对 Btrfs 文件系统运行的，则“设备上没有剩余空间”问题通常是由于系统上做为快照储存的数据过多所致。

您可以从 Snapper 中去除一些快照，不过，快照不会立即删除，可能不能释放您需要的空间容量。

若要从 Snapper 中删除文件：

1. 打开一个终端控制台。
2. 在命令提示符处，输入 **`btrfs filesystem show`**，例如：

```
> sudo btrfs filesystem show
Label: none uuid: 40123456-cb2c-4678-8b3d-d014d1c78c78
 Total devices 1 FS bytes used 20.00GB
 devid 1 size 20.00GB used 20.00GB path /dev/sda3
```

3. 输入

```
> sudo btrfs fi balance start MOUNTPOINT -dusage=5
```

   此命令会尝试将数据重新放置在空的或接近空的数据块中，从而允许收回空间并将其重新指派给元数据。此操作可能需要一些时间（1 TB 数据可能需要很多小时），不过，在此期间系统仍可以使用。

4. 列出快照程序中的快照。输入

```
> sudo snapper -c root list
```

5. 从 Snapper 中删除一或多个快照。输入

```
> sudo snapper -c root delete SNAPSHOT_NUMBER(S)
```

   务必先删除最旧的快照。快照生成的时间越长，其占用的空间就越大。

为了避免此问题发生，您可以更改 Snapper 清理算法。有关详细信息，请参见 《管理指南》，第 10 章 “通过 Snapper 进行系统恢复和快照管理”，第 10.6.1.2 节 “清理算法”。控制快照清理的配置值为 `EMPTY_*`、`NUMBER_*` 和 `TIMELINE_*`。

如果在文件系统磁盘上搭配使用 Snapper 和 Btrfs，建议您保留两倍于标准储存建议的磁盘空间容量。YaST 分区程序会自动在 Btrfs 储存建议中为根文件系统建议标准磁盘空间的两倍容量。

1.14.1.2 日志、崩溃和缓存文件使用的磁盘空间

如果系统磁盘填满了数据，您可以尝试从 `/var/log`、`/var/crash`、`/var/lib/systemd/coredump` 和 `/var/cache` 中删除文件。

Btrfs `root` 文件系统子卷 `/var/log`、`/var/crash` 和 `/var/cache` 在正常运作情况下可能会使用所有可用的磁盘空间，此时将导致系统故障。为避免出现此状况，SUSE Linux Enterprise Server 提供了 Btrfs 子卷配额支持。有关详细信息，请参见 第 1.2.5 节 “Btrfs 子卷定额支持”。

在测试和开发计算机上，尤其是当应用程序频繁崩溃时，您也可能想查看 `/var/lib/systemd/coredump`，内核转储就储存在其中。

1.14.2 Btrfs：跨设备平衡数据

`btrfs balance` 命令是 `btrfs-progs` 软件包的一部分。它可以在以下示例情况下平衡 Btrfs 文件系统上的块组：

- 假设您有一个 1 TB 驱动器，其中的 600 GB 被数据使用，然后又添加了另一个 1 TB 驱动器。理论上，平衡后将导致每个驱动器上各有 300 GB 的已用空间。
- 您的设备上有大量接近空的区块。在执行平衡清除这些区块之前，它们的空间都将不可用。
- 您需要根据其使用百分比压缩半空的块组。以下命令将平衡使用率等于或小于 5% 的块组：

  ```
  > sudo btrfs balance start -dusage=5 /
  ```

  提示

  `/usr/lib/systemd/system/btrfs-balance.timer` 计时器负责每月清理未使用的块组。

- 您需要清除块设备的未满部分，更均匀地分布数据。
- 您需要在不同的 RAID 类型之间迁移数据。例如，要将一组磁盘上的数据从 RAID1 转换到 RAID5，请运行以下命令：

```
> sudo btrfs balance start -dprofiles=raid1,convert=raid5 /
```

提示

要微调 Btrfs 文件系统上平衡数据的默认行为（例如，平衡的频率或挂载点），请检查并自定义 `/etc/sysconfig/btrfsmaintenance`。相关选项以 `BTRFS_BALANCE_` 开头。

有关 **`btrfs balance`** 命令用法的细节，请参见其手册页 (**`man 8 btrfs-balance`**)。

1.14.3 不要在 SSD 中进行碎片整理

Linux 文件系统包含相应的机制用于避免数据碎片，因此通常没有必要执行碎片整理。但在某些使用场合下，数据碎片不可避免，而对硬盘进行碎片整理可以明显提高性能。

这种做法仅适用于传统的硬盘。在使用闪存储存数据的固态硬盘 (SSD) 中，固件提供的算法可以确定要将数据写入哪些芯片。数据通常分散在设备的各个位置。因此，对 SSD 进行碎片整理并不能获得所需的效果，反而会因为写入不必要的数据而缩短 SSD 的寿命。

出于上述原因，SUSE 肯定地建议**不要**对 SSD 进行碎片整理。某些供应商还会警告对其固态硬盘进行碎片整理所产生的后果。这些品牌包括但不限于：

- HPE 3PAR StoreServ All-Flash
- HPE 3PAR StoreServ Converged Flash

1.15 更多信息

上面介绍的每个文件系统项目都有自己的主页，可以在其中找到邮件列表信息、更多文档和常见问题：

- Kernel.org 上的 Btrfs Wiki：https://btrfs.wiki.kernel.org/
- E2fsprogs: Ext2/3/4 File System Utilities: http://e2fsprogs.sourceforge.net/
- OCFS2 Project（OCFS2 项目）：https://oss.oracle.com/projects/ocfs2/

Wikipedia 项目上的“Comparison of File Systems”（文件系统比较，网址：http://en.wikipedia.org/wiki/Comparison_of_file_systems#Comparison）中提供了对各种文件系统（不仅仅是 Linux 文件系统）更深入的比较。

2 调整文件系统的大小

调整文件系统大小（不要与调整分区或卷大小混淆）可用于将物理卷上的空间变为可用状态，或使用物理卷上可用的其他空间。

2.1 使用案例

强烈建议您使用 YaST 分区程序来调整分区或逻辑卷的大小。这样一来，文件系统将自动调整为分区或卷的新大小。不过，在某些情况下，您需要手动调整文件系统的大小，因为 YaST 不支持它们：

- 调整虚拟机 Guest 的虚拟磁盘大小之后。
- 调整网络附加储存中的卷大小之后。
- 手动调整分区（例如通过使用 **`fdisk`** 或 **`parted`**）或逻辑卷（例如通过使用 **`lvresize`**）的大小之后。
- 要缩小 Btrfs 文件系统的大小时（从 SUSE Linux Enterprise Server 12 开始，YaST 仅支持增大 Btrfs 文件系统）。

2.2 调整大小指导原则

调整任何文件系统的大小都存在一定的风险，可能会造成数据遗失。

警告：备份数据

为了避免数据丢失，请确保在开始任何调整大小任务之前备份您的数据。

计划调整文件系统大小时，请考虑以下指导原则。

2.2.1 支持调整大小的文件系统

文件系统必须支持调整大小才能利用卷可用空间增加功能。SUSE Linux Enterprise Server 中提供了可用于文件系统 Ext2、Ext3 和 Ext4 的文件系统调整大小实用程序。这些实用程序支持如下增加和减小大小：

表 2.1： 文件系统对调整大小的支持

文件系统	实用程序	增加大小（增大）	减小大小（收缩）
Btrfs	`btrfs filesystem resize`	联机	联机
XFS	`xfs_growfs`	联机	不支持
Ext2	`resize2fs`	联机或脱机	仅限脱机
Ext3	`resize2fs`	联机或脱机	仅限脱机
Ext4	`resize2fs`	联机或脱机	仅限脱机

2.2.2 增加文件系统的大小

您可以将文件系统增大到设备上的最大可用空间，或指定一个准确大小。请确保在尝试增加文件系统的大小之前先增加设备或逻辑卷的大小。

为文件系统指定精确大小时，请确保新大小满足以下条件：

- 新大小必须大于现有数据的大小；否则会发生数据丢失。
- 新大小必须等于或小于当前设备大小，因为文件系统大小不能超出可用空间。

2.2.3 减小文件系统的大小

当减小设备上的文件系统的大小时，请确保新的大小满足以下条件：

- 新大小必须大于现有数据的大小；否则会发生数据丢失。
- 新大小必须等于或小于当前设备大小，因为文件系统大小不能超出可用空间。

如果还计划减小用于保存文件系统的逻辑卷的大小，请确保在尝试减小设备或逻辑卷的大小之前先减小文件系统的大小。

重要：XFS

XFS 格式文件系统的大小无法减少，因为 XFS 不支持此功能。

2.3 更改 Btrfs 文件系统的大小

挂载 Btrfs 文件系统后，您可以使用 **`btrfs filesystem resize`** 命令来更改该文件系统的大小。挂载了文件系统时，增加和缩小大小均受支持。

1. 打开一个终端控制台。
2. 确定您要更改的文件系统已挂载。
3. 使用 **`btrfs filesystem resize`** 命令通过下列其中一种方法更改文件系统的大小：
   - 要将文件系统大小扩展为设备的最大可用大小，请输入

     ```
     > sudo btrfs filesystem resize max /mnt
     ```

   - 要将文件系统扩展为指定大小，请输入

     ```
     > sudo btrfs filesystem resize SIZE /mnt
     ```

     将 `SIZE` 替换为所需大小（以字节为单位）。您还可以为值指定单位，例如 50000K (KB)、250M (MB) 或 2G (GB)。您也可以在值前面加上加号 (+) 或减号 (-)，分别指定增加或缩小至目前的大小：

     ```
     > sudo btrfs filesystem resize +SIZE /mnt
     sudo btrfs filesystem resize -SIZE /mnt
     ```

4. 通过输入以下命令，检查已挂载文件系统的调整大小的结果

```
> df -h
```

磁盘可用 (**df**) 命令显示磁盘的总大小、使用的块数以及文件系统上可用的块数。-h 选项会以可辨识的格式列印大小，如 1K、234M 或 2G。

2.4 更改 XFS 文件系统的大小

挂载 XFS 文件系统后，您可以使用 **xfs_growfs** 命令来增加该文件系统的大小。XFS 文件系统的大小无法减少。

1. 打开一个终端控制台。
2. 确定您要更改的文件系统已挂载。
3. 使用 **xfs_growfs** 命令增加文件系统的大小。下面的示例会将文件系统的大小扩充为最大可用值。请参见 **man 8 xfs_growfs** 了解更多选项。

   ```
   > sudo xfs_growfs -d /mnt
   ```

4. 通过输入以下命令，检查已挂载文件系统的调整大小的结果

   ```
   > df -h
   ```

   磁盘可用 (**df**) 命令显示磁盘的总大小、使用的块数以及文件系统上可用的块数。-h 选项会以可辨识的格式列印大小，如 1K、234M 或 2G。

2.5 更改 ext2、ext3 或 ext4 文件系统的大小

不管是否挂载了相应的分区，都可以使用 **resize2fs** 命令增加 Ext2、Ext3 和 Ext4 文件系统的大小。若要减少 Ext 文件系统的大小，需要将其卸载。

1. 打开一个终端控制台。
2. 如果应减少文件系统的大小，请将它卸载。
3. 使用下列方法之一更改文件系统的大小：

- 要将文件系统大小扩展为名为 `/dev/sda1` 的设备的最大可用大小，请输入

  ```
  > sudo resize2fs /dev/sda1
  ```

  如果未指定大小参数，大小将默认为该分区的大小。

- 若要将文件系统更改为特定大小，请输入

  ```
  > sudo resize2fs /dev/sda1 SIZE
  ```

  `SIZE` 参数指定为文件系统请求的新大小。如果不指定任何单位，则大小参数的单位是文件系统的块大小。也可以选择在大小参数后面加上下列其中一种单位指示项：`s` 表示 512 字节扇区；`K` 表示 KB（1 KB 为 1024 字节）；`M` 表示 MB；`G` 表示 GB。

等到调整大小完成再继续。

4. 如果未挂载该文件系统，则现在挂载它。

5. 通过输入以下命令，检查已挂载文件系统的调整大小的结果

   ```
   > df -h
   ```

   磁盘可用 (**df**) 命令显示磁盘的总大小、使用的块数以及文件系统上可用的块数。-h 选项会以可辨识的格式列印大小，如 1K、234M 或 2G。

3 挂载储存设备

本章概述在挂载设备期间会使用哪些设备标识符，并提供有关挂载网络储存的细节。

3.1 了解 UUID

UUID（全球唯一标识符）是表示文件系统的 128 位数字，在本地系统和其他系统中都是唯一的。它根据系统硬件信息和时戳（做为其种子的一部分）随机生成。UUID 通常用于唯一性标记设备。

使用非永久性的“传统”设备名称（例如 `/dev/sda1`）可能在添加储存时导致系统出现无法引导的情况。例如，如果将根 (`/`) 指派给 `/dev/sda1`，则在挂接 SAN 或将其他硬盘加入系统后，系统可能会将它重新指派给 `/dev/sdg1`。在此情况下，需要调整启动加载程序配置和 `/etc/fstab` 文件，否则系统将无法引导。

默认情况下，UUID 在引导加载程序以及引导设备的 `/etc/fstab` 文件中使用。UUID 是文件系统的一个属性，如果重新格式化驱动器，UUID 会更改。如果不想使用设备名称的 UUID，另一种替代方法是使用 ID 或标签识别设备。

您还可以将 UUID 用做组装与激活软件 RAID 设备的准则。当创建 RAID 时，**`md`** 驱动程序会为该设备生成一个 UUID，并将该值储存在 `md` 超块中。

您可以在 `/dev/disk/by-uuid` 目录中找到任何块设备的 UUID。UUID 项目示例如下所示：

```
> ls -og /dev/disk/by-uuid/
lrwxrwxrwx 1 10 Dec  5 07:48 e014e482-1c2d-4d09-84ec-61b3aefde77a -> ../../sda1
```

3.2 udev 的永久设备名称

从 Linux 内核 2.6 开始，**`udev`** 使用永久性设备命名方式，为动态的 `/dev` 目录提供了一种用户空间解决方案。作为热插拔系统的一部分，在系统中添加或删除设备时会执行 **`udev`**。

使用一个规则列表来针对特定设备属性进行匹配。**udev** 规则基础设施（在 /etc/udev/rules.d 目录中定义）为所有磁盘设备提供了稳定的名称，无论它们的识别顺序或该设备使用的连接如何。**udev** 工具检查内核创建的用来根据特定总线、驱动器类型或文件系统应用命名规则的每个相应块设备。有关如何为 **udev** 定义您自己的规则的信息，请参见 Writing udev Rules (http://reactivated.net/writing_udev_rules.html)（编写 udev 规则）。

和动态的内核提供的设备节点名一起，**udev** 还在 /dev/disk 目录中维护有指向该设备的永久符号链接的类，该目录进一步按 by-id、by-label、by-path 和 by-uuid 子目录分类。

注意：UUID 生成器

除了 **udev** 之外的其他程序（如 LVM 或 **md**）也可生成 UUID，但它们不在 /dev/disk 中列出。

有关使用 udev 来管理设备的详细信息，请参见《管理指南》, 第 29 章 “使用 udev 进行动态内核设备管理”。

有关 udev 命令的详细信息，请参见 **man 7 udev**。

3.3 挂载网络储存设备

对于某些类型的储存设备，需要为其配置网络并确保网络可用，然后 **systemd.mount** 才会开始挂载这些设备。要推迟这种设备的挂载，请将 _netdev 选项添加到每个特定网络储存设备的 /etc/fstab 文件中。示例如下：

```
mars.example.org:/nfsexport  /shared   nfs  defaults,_netdev    0  0
```

4 用于块设备操作的多层缓存

多层缓存是一种复制的/分布式缓存，它至少包括两个层：一个层由速度较慢但较为廉价的旋转块设备（硬盘）表示，另一个层成本更高，但执行数据操作的速度更快（例如，SSD 闪存盘）。

SUSE Linux Enterprise Server 为闪存设备与旋转设备的缓存实施两种不同的解决方案：`bcache` 和 `lvmcache`。

4.1 一般术语

本节对在介绍缓存相关功能时经常用到的几个术语进行了解释：

迁移

将逻辑块的主副本从一个设备移到另一个设备。

升级

从慢速设备迁移到快速设备。

降级

从快速设备迁移到慢速设备。

源设备

大型慢速块设备。它始终包含逻辑块的副本，该副本可能已过时或者与缓存设备上的副本保持同步（取决于策略）。

缓存设备

小型高速块设备。

元数据设备

一个小型设备，用于记录哪些块在缓存中、哪些块是脏的，以及供策略对象使用的附加提示。此信息可放在缓存设备上，但将它隔离可让卷管理器对它进行不同的配置，例如，配置为镜像以提高稳定性。元数据设备只能由单个缓存设备使用。

脏块

如果某个进程将信息写入缓存中的某个数据块，则该缓存的块将被标记为**脏**块，因为该块在缓存中已被覆盖，需要写回到原始设备。

缓存未命中

I/O 操作请求会先指向已缓存设备的缓存。如果找不到请求的值，则会在设备本身中查找，因此速度会变慢。这称为**缓存未命中**。

缓存命中

如果在已缓存设备的缓存中找到请求的值，则可以快速提供该值。这称为**缓存命中**。

冷缓存

不保存任何值（为空）且导致**缓存未命中**的缓存。在执行已缓存块设备的操作过程中，冷缓存中会填充数据，从而变为**暖**缓存。

暖缓存

已保存了一些值并且可能会导致**缓存命中**的缓存。

4.2 缓存模式

下面是多层缓存使用的基本缓存模式：**写回**、**直写**、**绕写**和**直通**。

写回

写入已缓存块的数据只会存入缓存，并且该块将标记为脏块。这是默认的缓存模式。

直写

只有在同时命中源设备和缓存设备之后，向已缓存块的写入才会完成。在**直写**缓存中，干净块将保持干净状态。

绕写

类似于直写缓存的一种技术，不过，写 I/O 将直接写入永久性储存，并绕过缓存。这可以防止缓存因写 I/O 而填满，导致以后不可重新读取，不过，缺点是对最近写入数据的读取请求会造成“缓存未命中”，因而需要从慢速大容量储存中读取这些数据，致使发生较高延迟。

直通

要启用**直通**模式，缓存必须是干净的。将绕过缓存，从源设备为读取请求提供服务。写请求将转到源设备，使缓存块“失效”。**直通**允许您激活缓存设备时不必考虑数据一致性，而这是可以维护的。随着写操作的不断进行，缓存将逐渐变为冷状态。如果您以后可以校

验缓存的一致性，或者可以使用 `invalidate_cblocks` 消息来建立这种一致性，则可以在缓存设备仍处于暖状态时，将它切换到**直写**或**写回**模式。或者，可以在切换到所需的缓存模式之前，先丢弃缓存内容。

4.3 bcache

`bcache` 是一个 Linux 内核块层缓存。它允许使用一个或多个高速磁盘驱动器（例如 SSD）作为一个或多个速度低得多的硬盘的缓存。`bcache` 支持直写和写回，不受所用文件系统的约束。默认情况下，它只缓存随机读取和写入，这也是 SSD 的强项。它还适合用于台式机、服务器和高端储存阵列。

4.3.1 主要功能

- 可以使用单个缓存设备来缓存任意数量的后备设备。在运行时可以挂接和分离已挂载及使用中的后备设备。
- 在非正常关机后恢复 - 只有在缓存与后备设备一致后才完成写入。
- SSD 拥塞时限制传至 SSD 的流量。
- 高效的写回实施方案。脏数据始终按排序顺序写出。
- 稳定可靠，可在生产环境中使用。

4.3.2 设置 bcache 设备

本节介绍设置和管理 `bcache` 设备的步骤。

1. 安装 `bcache-tools` 软件包：

```
> sudo zypper in bcache-tools
```

2. 创建后备设备（通常是一个机械驱动器）。后备设备可以是整个设备、一个分区或任何其他标准块设备。

```
> sudo make-bcache -B /dev/sdb
```

3. 创建缓存设备（通常是一个 SSD 磁盘）。

```
> sudo make-bcache -C /dev/sdc
```

本示例使用了默认的块大小和储存桶大小，分别为 512 B 和 128 KB。块大小应与后备设备的扇区大小（通常为 512 或 4k）匹配。储存桶大小应与缓存设备的擦除块大小匹配，以便减少写入放大现象。例如，如果使用具有 4k 扇区的硬盘和具有 2 MB 擦除块大小的 SSD，则此命令将如下所示：

```
sudo make-bcache --block 4k --bucket 2M -C /dev/sdc
```

提示：多设备支持

make-bcache 可同时准备和注册多个后备设备与一个缓存设备。在这种情况下，以后您不需要将缓存设备手动挂接到后备设备：

```
> sudo make-bcache -B /dev/sda /dev/sdb -C /dev/sdc
```

4. bcache 设备将显示为

```
/dev/bcacheN
```

和

```
/dev/bcache/by-uuid/UUID
/dev/bcache/by-label/LABEL
```

您可以像平时一样正常格式化和挂载 bcache 设备：

```
> sudo mkfs.ext4 /dev/bcache0
> sudo mount /dev/bcache0 /mnt
```

您可以在 /sys/block/bcacheN/bcache 中通过 sysfs 控制 bcache 设备。

5. 注册缓存设备和后备设备后，需要将后备设备挂接到相关的缓存集才能启用缓存：

```
> echo CACHE_SET_UUID > /sys/block/bcache0/bcache/attach
```

其中，CACHE_SET_UUID 可在 /sys/fs/bcache 中找到。

6. 默认情况下，bcache 使用直通缓存模式。要更改模式，例如，更改为写回模式，请运行

```
> echo writeback > /sys/block/bcache0/bcache/cache_mode
```

4.3.3 使用 sysfs 配置 bcache

bcache 设备使用 sysfs 接口来储存其运行时配置值。这样，您便可以更改 bcache 后备设备和缓存磁盘的行为，或查看其使用统计数字。

有关 bcache sysfs 参数的完整列表，请查看 /usr/src/linux/Documentation/bcache.txt 文件的内容，主要查看 SYSFS - BACKING DEVICE、SYSFS - BACKING DEVICE STATS 和 SYSFS - CACHE DEVICE 部分。

4.4 lvmcache

lvmcache 是由逻辑卷 (LV) 组成的缓存机制。它使用 dm-cache 内核驱动程序，支持直写（默认）和写回缓存模式。lvmcache 可将大型慢速 LV 的部分数据动态迁移到更快、更小的 LV，从而提高其性能。有关 LVM 的详细信息，请参见第 II 部分 “逻辑卷 (LVM)”。

LVM 将小型快速 LV 称为**缓存池 LV**。大型慢速 LV 称为**源 LV**。由于 dm-cache 的要求，LVM 进一步将缓存池 LV 分割成两个设备：**缓存数据 LV** 和**缓存元数据 LV**。来自源 LV 的数据块副本保存在缓存数据 LV 中，以提高速度。缓存元数据 LV 保存记帐信息，这些信息指定数据块的储存位置。

4.4.1 配置 lvmcache

本节介绍创建和配置基于 LVM 的缓存的步骤。

1. **创建源 LV。**创建新 LV，或使用现有 LV 作为源 LV：

```
> sudo lvcreate -n ORIGIN_LV -L 100G vg /dev/SLOW_DEV
```

2. **创建缓存数据 LV。**此 LV 将保存来自源 LV 的数据块。此 LV 的大小是缓存的大小，将报告为缓存池 LV 的大小。

```
> sudo lvcreate -n CACHE_DATA_LV -L 10G vg /dev/FAST
```

3. **创建缓存元数据 LV。**此 LV 将保存缓存池元数据。此 LV 的大小应该比缓存数据 LV 大约小 1000 倍，其最小大小为 8MB。

```
> sudo lvcreate -n CACHE_METADATA_LV -L 12M vg /dev/FAST
```

列出您目前为止所创建的卷：

```
> sudo lvs -a vg
LV                 VG   Attr        LSize   Pool Origin
cache_data_lv      vg   -wi-a-----  10.00g
cache_metadata_lv vg   -wi-a-----  12.00m
origin_lv          vg   -wi-a----- 100.00g
```

4. **创建缓存池 LV。**将数据 LV 和元数据 LV 组合成一个缓存池 LV。同时还可以设置缓存池 LV 的行为。

   `CACHE_POOL_LV` 与 `CACHE_DATA_LV` 同名。

   `CACHE_DATA_LV` 重命名为 `CACHE_DATA_LV`_cdata，并且会隐藏起来。

   `CACHE_META_LV` 重命名为 `CACHE_DATA_LV`_cmeta，并且会隐藏起来。

```
> sudo lvconvert --type cache-pool \
 --poolmetadata vg/cache_metadata_lv vg/cache_data_lv
```

```
> sudo lvs -a vg
LV                     VG   Attr       LSize   Pool Origin
cache_data_lv          vg   Cwi---C---  10.00g
[cache_data_lv_cdata]  vg   Cwi-------  10.00g
[cache_data_lv_cmeta]  vg   ewi-------  12.00m
origin_lv              vg   -wi-a----- 100.00g
```

5. **创建缓存 LV。**通过将缓存池 LV 链接到源 LV 来创建缓存 LV。

用户可访问的缓存 LV 与源 LV 同名，源 LV 将变成重命名为 ORIGIN_LV_corig 的隐藏 LV。
CacheLV 与 ORIGIN_LV 同名。
ORIGIN_LV 重命名为 ORIGIN_LV_corig，并且会隐藏起来。

```
> sudo lvconvert --type cache --cachepool vg/cache_data_lv vg/origin_lv
```

```
> sudo lvs -a vg
LV              VG   Attr       LSize   Pool   Origin
cache_data_lv          vg   Cwi---C---  10.00g
[cache_data_lv_cdata]  vg   Cwi-ao----  10.00g
[cache_data_lv_cmeta]  vg   ewi-ao----  12.00m
origin_lv              vg   Cwi-a-C--- 100.00g cache_data_lv
 [origin_lv_corig]
[origin_lv_corig]      vg   -wi-ao---- 100.00g
```

4.4.2 去除缓存池

可通过多种方法关闭 LV 缓存。

4.4.2.1 从缓存 LV 分离缓存池 LV

您可以从缓存 LV 断开与缓存池 LV 的连接，留下一个未使用的缓存池 LV 和一个未缓存的源 LV。数据将根据需要从缓存池写回到源 LV。

```
> sudo lvconvert --splitcache vg/origin_lv
```

4.4.2.2 去除缓存池 LV 但不去除其源 LV

以下命令会根据需要将数据从缓存池写回到源 LV，然后去除缓存池 LV，留下未缓存的源 LV。

```
> sudo lvremove vg/cache_data_lv
```

也可以使用以下替代命令从缓存 LV 断开与缓存池的连接，并删除缓存池：

```
> sudo lvconvert --uncache vg/origin_lv
```

4.4.2.3 去除源 LV 和缓存池 LV

去除缓存 LV 会同时去除源 LV 和链接的缓存池 LV。

```
> sudo lvremove vg/origin_lv
```

4.4.2.4 更多信息

可以在 `lvmcache` 手册页 (`lvmcache`man 7 lvmcache**）中找到有关** 的更多主题，例如支持的缓存模式、冗余的子逻辑卷、缓存策略，或者将现有 LV 转换为缓存类型。

II 逻辑卷 (LVM)

5 LVM 配置 54

6 LVM 卷快照 82

5 LVM 配置

本章介绍逻辑卷管理器 (LVM) 的原理，以及令其在许多情况下都能发挥效用的基本功能。YaST LVM 配置可以通过 YaST 专家分区程序完成。此分区工具用于编辑和删除现有分区并创建用于 LVM 的新分区。

警告：风险

使用 LVM 可能会增加一些风险，例如数据丢失。这些风险还包括应用程序崩溃、电源故障及有问题的命令。在实施 LVM 或重配置卷前，请保存数据。决不要在没有备份的情况下工作。

5.1 了解逻辑卷管理器

LVM 支持在多个物理卷（硬盘、分区、LUN）之间弹性分配硬盘空间。开发逻辑卷管理器是因为可能只有在安装期间初始分区完成后才需要更改硬盘空间的分段。因为在正在运行的系统中修改分区比较困难，LVM 提供了储存空间的虚拟池（卷组或 VG），如果需要，可以从中生成逻辑卷 (LV)。操作系统访问这些逻辑卷而不是物理分区。卷组可以跨多个磁盘，这样多个磁盘或部分磁盘可以构成一个 VG。LVM 以这种方式提供了一种对物理磁盘空间的抽象，从而能够以比物理分区更方便、更安全的方式更改硬盘空间的分段。

图 5.1 “物理分区与 LVM”比较物理分区（左）和 lvm 分段（右）。在左侧，将一个磁盘分成 3 个物理分区 (PART)，每个分区指派了一个挂载点 (MP)，以便操作系统可以访问它们。在右侧，有两个磁盘，一个磁盘分为 2 个物理分区，另一个磁盘分为 3 个物理分区。定义了两个 LVM 卷组（VG1 和 VG2）。VG 1 包含磁盘 1 的两个分区和磁盘 2 的一个分区。VG 2 包含磁盘 2 的其余两个分区。

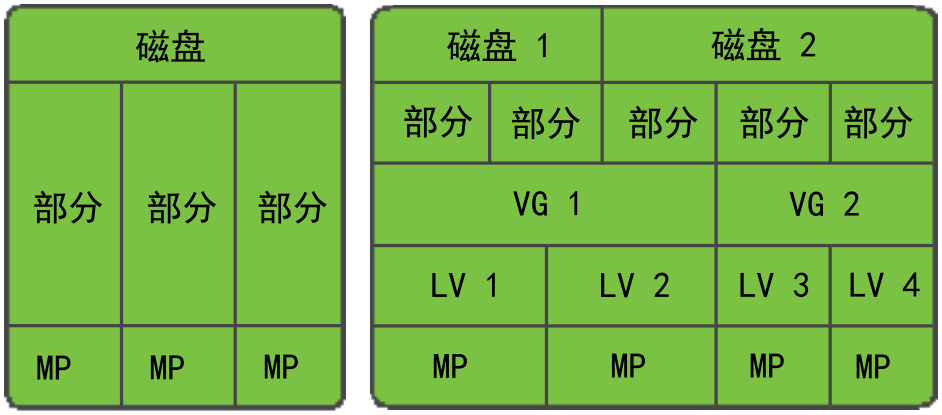


图 5.1： 物理分区与 LVM

在 LVM 中，合并到卷组的物理磁盘分区称为物理卷 (PV)。在图 5.1 “物理分区与 LVM” 的卷组中，定义了四个逻辑卷（LV 1 至 LV 4），操作系统可以通过关联的挂载点 (MP) 来使用这些逻辑卷。不同逻辑卷之间的边界不一定是任何分区边界。请参见本示例中 LV 1 和 LV 2 之间的边界。

LVM 功能：

- 可以将多块硬盘或多个分区合并为一个较大的逻辑卷。
- 如果配置合适，当可用空间用完后，可以扩大 LV（例如 `/usr`）。
- 通过使用 LVM，可以在正在运行的系统中添加硬盘或 LV。但这需要支持此类操作的可热插拔的硬件。
- 可激活**分段方式**，此方式将通过若干物理卷来分发逻辑卷的数据流。如果这些物理卷位于不同的磁盘上，则可提高读写性能（类似于 RAID 0）。
- 使用快照功能可以在正在运行的系统中执行一致的备份（尤其适合服务器）。

注意：LVM 和 RAID

即使 LVM 也支持 RAID 级别 0、1、4、5 和 6，我们仍建议您使用 `mdraid`（请参见第 7 章 “软件 RAID 配置”）。不过，LVM 可以与 RAID 0 和 1 搭配使用，因为 RAID 0 类似于通用逻辑卷管理（各个逻辑块将映射到物理设备上的块）。在 RAID 1 基础上使用的 LVM 可以跟踪镜像同步，并完全能够管理同步过程。使用更高的 RAID 级别时，需

要借助一个管理守护程序来监视挂接的磁盘的状态，并在磁盘阵列出现问题时通知管理员。LVM 包含此类守护程序，但在异常情况下（例如设备故障），该守护程序无法正常工作。

警告：IBM Z：LVM 根文件系统

如果您在 LVM 或软件 RAID 阵列中为系统配置了根文件系统，则必须将 `/boot` 置于单独的非 LVM 或非 RAID 分区上，否则系统将无法引导。此类分区的建议大小为 500 MB，建议的文件系统为 Ext4。

通过这些功能，使用 LVM 还对频繁使用的家用 PC 或小型服务器有用。如果您的数据储存量（如数据库、音乐存档或用户目录）不断增长，则 LVM 尤其有用。它支持您使用大于物理硬盘的文件系统。但是，请记住，使用 LVM 与使用传统的分区截然不同。

您可以使用 YaST 分区程序管理新的或现有的 LVM 储存对象。有关配置 LVM 的指示及详细信息，请参见官方的 LVM HOWTO (http://tldp.org/HOWTO/LVM-HOWTO/)。

5.2 创建卷组

LVM 卷组 (VG) 会将 Linux LVM 分区组织到一个逻辑空间池中。您可以从组的可用空间中划分出逻辑卷。组中的 Linux LVM 分区可以在相同或不同磁盘上。您可以添加分区或整个磁盘来扩大组的大小。

要使用整个磁盘，该磁盘不得包含任何分区。使用分区时，不得挂载分区。在将分区添加到 VG 时，YaST 会自动将它们的分区类型更改为 `0x8E Linux LVM`。

1. 启动 YaST 并打开分区程序。
2. 如果您需要重新配置现有分区设置，请执行下面操作。有关细节，请参考《部署指南》，第 10 章 “专家分区程序”，第 10.1 节 “使用专家分区程序”。如果您只想使用未使用的磁盘或已存在的分区，请跳过此步骤。

   ### 警告：未分区磁盘上的物理卷

   可以使用某个未分区的磁盘作为物理卷 (PV)，前提是该磁盘**不**是安装操作系统的磁盘，也不是操作系统从中引导的磁盘。

由于未分区的磁盘在系统级别显示为**未使用**，因此很容易被覆盖或被不正确地访问。

a. 要使用已包含分区的整个硬盘，请删除该磁盘上的所有分区。

b. 要使用当前已挂载的分区，请将其卸载。

3. 在左侧面板中，选择卷管理。
   右侧面板中即会打开现有卷组的列表。

4. 在“卷管理”页的左下角，单击添加卷组。

5. 按如下所示定义卷组：

   a. 指定卷组名称。
      如果在安装时创建卷组，建议对将包含 SUSE Linux Enterprise Server 文件系统的卷组采用名称 `system`。

   b. 指定物理区域大小。
      物理区域大小定义卷组中物理块的大小。卷组中的所有磁盘空间都是按此大小的区块来处理的。值可以是 1 KB 到 16 GB（2 的幂数形式）。通常将此值设置为 4 MB。

在 LVM1 中，4 MB 物理区域允许的最大 LV 大小为 256 GB，因为它仅支持每个 LV 最多 65534 个区域。SUSE Linux Enterprise Server 上使用的 LVM2 不会限制物理区域的数量。区域数过多不会影响逻辑卷的 I/O 性能，但会降低 LVM 工具的速度。

重要：物理区域大小

单个 VG 中不应混合有不同的物理区域大小。区域在初始设置后不应修改。

c. 在可用物理卷列表中，选择要成为此卷组一部分的 Linux LVM 分区，然后单击添加将它们移动到所选物理卷列表。

d. 单击完成。

新组将出现在卷组列表中。

6. 在“卷管理”页上，单击下一步，验证是否列出了新卷组，然后单击完成。

7. 要检查哪些物理设备属于卷组的一部分，可随时在运行的系统中打开 YaST 分区程序，然后单击卷管理 › 编辑 › 物理设备。单击中止离开此屏幕。

图 5.2： 名为 DATA 的卷组中的物理卷

5.3 创建逻辑卷

逻辑卷提供一个空间池，此空间池与硬盘提供的空间池类似。要让此空间可用，需要定义逻辑卷。逻辑卷类似于一个普通分区 - 您可以进行格式化并将其挂载系统。

使用 YaST 分区程序从现有卷组创建逻辑卷。请为每个卷组至少指派一个逻辑卷。您可以根据需要创建新的逻辑卷，直到卷组中的所有可用空间都用完为止。可以选择性地精简配置一个 LVM 逻辑卷，以便创建大小超出可用空间的逻辑卷（有关详细信息，请参见第 5.3.1 节 “精简配置的逻辑卷”）。

- **普通卷：** （默认）系统会立即分配卷的空间。
- **瘦池：** 逻辑卷是预留供瘦卷使用的空间池。瘦卷可以按需从瘦池分配它们所需的空间。
- **瘦卷：** 该卷会被创建为稀疏卷。瘦卷会按需从瘦池分配所需的空间。
- **镜像卷：** 创建的卷中包含定义数目的镜像。

过程 5.1： 设置逻辑卷

1. 启动 YaST 并打开分区程序。
2. 在左侧面板中，选择卷管理。右侧面板中即会打开现有卷组的列表。
3. 选择要在其中创建卷的卷组，然后选择逻辑卷 › 添加逻辑卷。
4. 提供卷的名称，然后选择普通卷（有关设置精简配置卷的相关信息，请参见第 5.3.1 节 “精简配置的逻辑卷”）。单击下一步继续。

5. 指定卷的大小和是否使用多个分段。

   使用带区卷时，将在多个物理卷之间分配数据。如果这些物理卷驻留在不同的硬盘上，则通常会提高读写性能（与 RAID 0 类似）。可用带区卷的最大数量为物理卷的数量。默认值 (1) 表示不使用多个带区卷。

6. 选择卷的角色。您在此处所做选择只会影响将要打开的对话框的默认值。这些值可在下一个步骤中更改。如果不确定，请选择原始卷（未格式化）。

7. 在格式化选项下，选择格式化分区，然后选择文件系统。选项菜单的内容取决于文件系统。通常不需要更改默认值。
   在挂载选项下，选择挂载分区，然后选择挂载点。单击 Fstab 选项，为卷添加特殊挂载选项。
8. 单击完成。
9. 单击下一步，校验是否列出了更改，然后单击完成。

5.3.1 精简配置的逻辑卷

LVM 逻辑卷可选择进行精简配置。精简配置可让您创建大小超出可用空间的逻辑卷。您创建包含未使用空间（预留供任意数目的瘦卷使用）的瘦池。瘦卷会被创建为稀疏卷，并会根据需要从瘦池分配空间。瘦池可以根据需要动态扩大，以实现储存空间的高效分配。精简配置的卷还支持快照（可以使用 Snapper 进行管理）— 有关详细信息，请参见《管理指南》, 第 10 章 “通过 Snapper 进行系统恢复和快照管理”。

要设置精简配置的逻辑卷，请按过程 5.1 “设置逻辑卷” 中所述步骤操作。选择卷类型时，不要选择普通卷，而要选择瘦卷或瘦池。

瘦池

逻辑卷是预留供瘦卷使用的空间池。瘦卷可以按需从瘦池分配它们所需的空间。

瘦卷

该卷会被创建为稀疏卷。瘦卷会按需从瘦池分配所需的空间。

重要：群集中精简配置的卷

要使用群集中的精简配置卷，使用它的瘦池和瘦卷必须在单个群集资源中管理。如此可使瘦卷和瘦池始终在同一个节点上以独占方式挂载。

5.3.2 创建镜像卷

可以创建包含多个镜像的逻辑卷。LVM 会确保将写入底层物理卷的数据镜像到不同的物理卷。因此，即使某个物理卷崩溃，您仍可访问逻辑卷上的数据。LVM 还会保留一个日志文件用于管理同步过程。日志中包含有关哪些卷区域当前正在与镜像同步的信息。默认情况下，日志储存在磁盘上，当情况允许时，会储存在与镜像不同的磁盘上。不过，您可以为日志指定一个不同的位置，例如，指定易失性内存。

目前可以使用两种类型的镜像实施：“正常”的（非 raid）镜像逻辑卷和 `raid1` 逻辑卷。

创建镜像逻辑卷后，可对这些卷执行标准操作，例如激活、扩展和去除。

5.3.2.1 设置镜像非 RAID 逻辑卷

要创建镜像卷，请使用 **`lvcreate`** 命令。下面的示例将创建一个 500 GB 的逻辑卷，其中包含两个名为 **lv1** 的镜像，并使用卷组 **vg1**。

```
> sudo lvcreate -L 500G -m 2 -n lv1 vg1
```

此类逻辑卷是一种线性卷（无分段），可提供文件系统的三个副本。`m` 选项指定镜像的计数。`L` 选项指定逻辑卷的大小。

逻辑卷会划分为几个默认大小为 512 KB 的区域。如果需要不同大小的区域，请使用 `-R` 选项并后接所需的区域大小（以 MB 为单位）。或者，可以在 `lvm.conf` 文件中编辑 `mirror_region_size` 选项来配置首选的区域大小。

5.3.2.2 设置 raid1 逻辑卷

由于 LVM 支持 RAID，您可以使用 RAID1 来实施镜像。与非 raid 镜像相比，这种实施具有以下优点：

- LVM 会为每个镜像映像维护一个完全冗余的位图区，从而提高了其故障处理能力。
- 可以暂时从阵列中分割出镜像映像，然后将它们重新合并。
- 阵列可以处理暂时性故障。
- LVM RAID 1 实施支持快照。

但在另一方面，这种类型的镜像实施不允许在群集卷组中创建逻辑卷。

要使用 RAID 创建镜像卷，请发出以下命令：

```
> sudo lvcreate --type raid1 -m 1 -L 1G -n lv1 vg1
```

其中的各选项/参数的含义如下：

- `--type` - 需要指定 `raid1`，否则该命令将使用隐式分段类型 `mirror` 并创建非 raid 镜像。
- `-m` - 指定镜像的计数。
- `-L` - 指定逻辑卷的大小。
- `-n` - 使用此选项指定逻辑卷的名称。
- `vg1` - 逻辑卷使用的卷组的名称。

LVM 将为阵列中的每个数据卷创建一个区域大小的逻辑卷。如果您有两个镜像卷，LVM 将另外创建两个卷用于储存元数据。

创建 RAID 逻辑卷后，您可以像使用普通的逻辑卷一样使用该卷。您可以将它激活、扩展，等等

5.4 自动激活非根 LVM 卷组

非根 LVM 卷组的激活行为由 `/etc/lvm/lvm.conf` 文件控制，同时由 `auto_activation_volume_list` 参数控制。默认情况下，该参数为空，也就是说，会激活所有卷。如果只想激活某些卷组，请将名称括在引号中，并用逗号分隔各名称，例如：

```
auto_activation_volume_list = [ "vg1", "vg2/lvol1", "@tag1", "@*" ]
```

如果您已在 `auto_activation_volume_list` 参数中定义了一个列表，则会发生以下情况：

1. 首先会根据此列表检查每个逻辑卷。
2. 如果两者不匹配，则不激活该逻辑卷。

默认情况下，在 dracut 重启动系统时，非根 LVM 卷组会自动激活。您可以使用此参数在系统重启时激活所有卷组，或仅激活指定的非根 LVM 卷组。

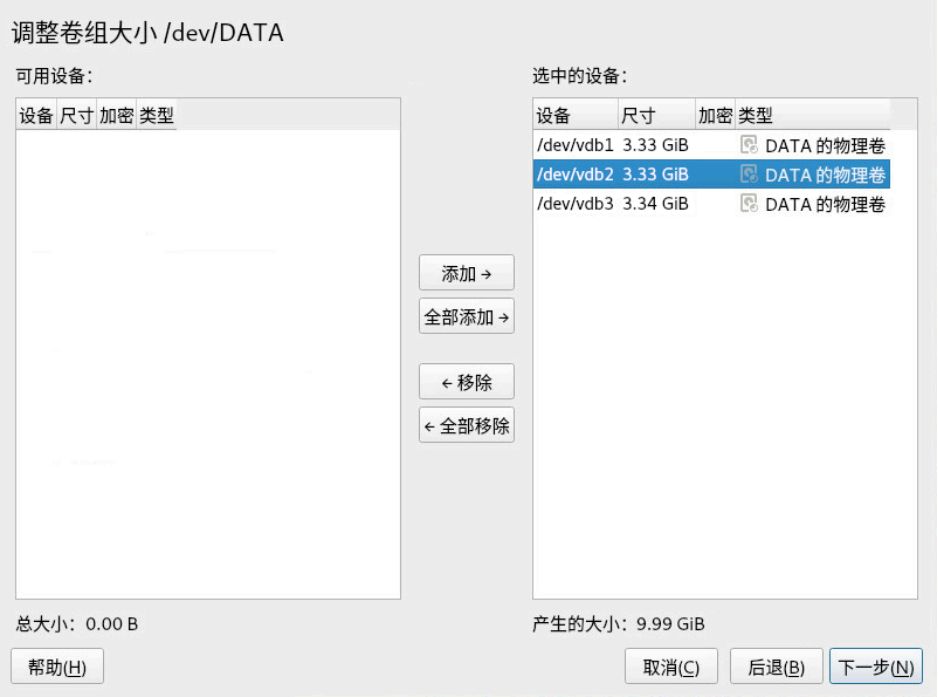
5.5 调整现有卷组的大小

您随时都可在正在运行的系统中添加物理卷来扩展卷组提供的空间，而不会导致服务中断。这可让您将逻辑卷添加到组中，或如第 5.6 节 “调整逻辑卷的大小” 中所述扩大现有卷的大小。

还可以通过删除物理卷来缩小卷组的大小。YaST 只允许删除当前未使用的物理卷。要了解哪些物理卷当前正在使用，请运行下列命令。PE Ranges 列中列出的分区（物理卷）即是使用中的分区：

```
> sudo pvs -o vg_name,lv_name,pv_name,seg_pe_ranges
root's password:
  VG   LV    PV         PE Ranges
             /dev/sda1
  DATA DEVEL /dev/sda5  /dev/sda5:0-3839
  DATA       /dev/sda5
  DATA LOCAL /dev/sda6  /dev/sda6:0-2559
  DATA       /dev/sda7
  DATA       /dev/sdb1
  DATA       /dev/sdc1
```

1. 启动 YaST 并打开分区程序。
2. 在左侧面板中，选择卷管理。右侧面板中即会打开现有卷组的列表。
3. 选择要更改的卷组，激活物理卷选项卡，然后单击更改。



4. 执行以下操作之一：

   - **添加：** 要扩展卷组大小，请将一个或多个物理卷（LVM 分区）从可用物理卷列表移动到所选物理卷列表。
   - **去除：** 要缩小卷组的大小，可将一个或多个物理卷（LVM 分区）从所选物理卷列表移到可用物理卷列表。

5. 单击完成。

6. 单击下一步，校验是否列出了更改，然后单击完成。

5.6 调整逻辑卷的大小

如果卷组中有未使用的可用空间，则您可以增大逻辑卷以提供更多可用空间。您还可以将卷的大小减少为其他逻辑卷可使用的卷组的可用空间。

注意：“联机”调整大小

减少卷的大小时，YaST 会同时自动调整其文件系统的大小。当前已挂载的卷是否可以“联机”（即保持已挂载状态）调整大小取决于其文件系统。Btrfs、XFS、Ext3 和 Ext4 支持联机扩展文件系统。

仅 Btrfs 支持联机缩小文件系统。要缩小 Ext2/3/4 文件系统，需要将其卸载。无法缩小以 XFS 格式化的卷，因为 XFS 不支持文件系统缩小。

1. 启动 YaST 并打开分区程序。
2. 在左侧面板中，选择卷管理。右侧面板中即会打开现有卷组的列表。
3. 选择要更改的逻辑卷，然后单击调整大小。

4. 使用下列选项之一设置所需大小：
   - **最大大小：** 扩大逻辑卷的大小以使用卷组中的所有剩余空间。
   - **最小大小：** 将逻辑卷的大小减少到数据和文件系统元数据所占用的大小。
   - **自定义大小：** 指定卷的新大小。此值必须介于上面列出的最小值和最大值之间。使用 K、M、G、T 表示 KB、MB、GB 和 TB（例如 `20G`）。

5. 单击确定。
6. 单击下一步，校验是否列出了更改，然后单击完成。

5.7 删除卷组或逻辑卷

警告：数据丢失

删除卷组会清空其每个成员分区中的所有数据。删除逻辑卷会损坏该卷上储存的所有数据。

1. 启动 YaST 并打开分区程序。
2. 在左侧面板中，选择卷管理。右侧面板中即会打开现有卷组的列表。
3. 选择要删除的卷组或逻辑卷，然后单击删除。
4. 根据您的选择，系统会显示警告对话框。单击是以确认对话框。
5. 单击下一步，校验是否列出了已删除卷组（删除项以红色字体表示），然后单击完成。

5.8 使用 LVM 命令

有关使用 LVM 命令的信息，请参见下表中所述的命令的**手册**页。所有命令都需要拥有 `root` 权限才能执行。请使用 **`sudo`** `COMMAND`（建议采用此方式），或者直接以 `root` 身份执行这些命令。

LVM 命令

`pvcreate` `DEVICE`

初始化设备（例如 `/dev/sdb1`），供 LVM 用作物理卷。如果指定的设备上有任何文件系统，将出现警告。请记住，仅当已安装 **`blkid`** 时（默认已安装），**`pvcreate`** 才会检查现有文件系统。如果 **`blkid`** 不可用，**`pvcreate`** 不会生成任何警告，因此您可能会丢失文件系统却看不到任何警告。

pvdisplay DEVICE

显示 LVM 物理卷的相关信息，例如当前它是否正在逻辑卷中使用。

vgcreate -c y VG_NAME DEV1 **[**DEV2**...]**

使用一个或多个指定的设备创建群集卷组。

vgcreate --activationmode ACTIVATION_MODE VG_NAME

配置卷组激活模式。可以指定下列值之一：

- complete - 只能激活不受缺失物理卷影响的逻辑卷，即使特定的逻辑卷能够容许这种故障也如此。
- degraded - 默认的激活模式。如果有足够的冗余级别来激活某个逻辑卷，则即使缺少某些物理卷，也能激活该逻辑卷。
- partial - 即使缺少某些物理卷，LVM 也会尝试激活卷组。如果某个非冗余逻辑卷缺少重要的物理卷，则通常无法激活该逻辑卷，并会将它作为错误目标进行处理。

vgchange -a [ey|n] VG_NAME

激活 (-a ey) 或停用 (-a n) 卷组及其逻辑卷的输入/输出。

激活群集中的某个卷时，请务必使用 ey 选项。此选项默认在装载脚本中使用。

vgremove VG_NAME

删除卷组。请在使用此命令之前删除逻辑卷，然后停用卷组。

vgdisplay VG_NAME

显示指定卷组的相关信息。

要了解卷组的物理区域总大小，请输入

```
> vgdisplay VG_NAME | grep "Total PE"
```

lvcreate -L SIZE **-n** LV_NAME VG_NAME

创建具有指定大小的逻辑卷。

lvcreate -L SIZE **--thinpool** POOL_NAME VG_NAME

基于卷组 VG_NAME 创建具有指定大小的瘦池 myPool。

下面的示例会从卷组 LOCAL 创建大小为 5 GB 的瘦池：

```
> sudo lvcreate -L 5G --thinpool myPool LOCAL
```

lvcreate -T VG_NAME/POOL_NAME **-V** SIZE **-n** LV_NAME

在池 POOL_NAME 中创建瘦逻辑卷。下面的示例会从卷组 LOCAL 上的池 myPool 创建名为 myThin1 的 1GB 瘦卷：

```
> sudo lvcreate -T LOCAL/myPool -V 1G -n myThin1
```

lvcreate -T VG_NAME/POOL_NAME **-V** SIZE **-L** SIZE **-n** LV_NAME

还可以通过一条命令同时创建瘦池和瘦逻辑卷：

```
> sudo lvcreate -T LOCAL/myPool -V 1G -L 5G -n myThin1
```

lvcreate --activationmode ACTIVATION_MODE LV_NAME

配置逻辑卷激活模式。可以指定下列值之一：

- complete - 仅当逻辑卷的所有物理卷都处于活动状态时才能激活该逻辑卷。
- degraded - 默认的激活模式。如果有足够的冗余级别来激活某个逻辑卷，则即使缺少某些物理卷，也能激活该逻辑卷。
- partial - 即使缺少某些物理卷，LVM 也会尝试激活卷。如果逻辑卷有一部分不可用，使用此选项可能会导致数据丢失。通常不使用此选项，但在恢复数据时，它可能会有用。

您可以通过在 /etc/lvm/lvm.conf 中指定 activation_mode 配置选项的上述值之一，来指定激活模式。

lvcreate -s [-L SIZE**] -n** SNAP_VOLUME SOURCE_VOLUME_PATH VG_NAME

创建指定逻辑卷的快照卷。如果未包含大小选项（-L 或 --size），则快照会被创建为瘦快照。

lvremove /dev/VG_NAME/LV_NAME

删除逻辑卷。

使用此命令之前，请先使用 **umount** 命令卸载逻辑卷以将其关闭。

lvremove SNAP_VOLUME_PATH

删除快照卷。

lvconvert --merge SNAP_VOLUME_PATH

将逻辑卷还原为快照的版本。

vgextend VG_NAME DEVICE

将指定的设备（物理卷）添加到现有卷组。

vgreduce VG_NAME DEVICE

从现有卷组中删除指定的物理卷。

确保物理卷当前未被逻辑卷使用。如果正在使用中，则必须使用 **pvmove** 命令将数据移至另一个物理卷。

lvextend -L SIZE **/dev/**VG_NAME**/**LV_NAME

扩大指定逻辑卷的大小。随后，您还必须相应地扩大文件系统以充分利用新扩充的空间。有关详细信息，请参见 第 2 章 “调整文件系统的大小”。

lvreduce -L SIZE **/dev/**VG_NAME**/**LV_NAME

减少指定逻辑卷的大小。

在缩小卷之前，请确保先减少文件系统的大小，否则可能会丢失数据。有关详细信息，请参见 第 2 章 “调整文件系统的大小”。

lvrename /dev/VG_NAME**/**LV_NAME **/dev/**VG_NAME**/**NEW_LV_NAME

重命名现有 LVM 逻辑卷。此操作不会更改卷组名。

提示：创建卷时绕过 udev

如果您想使用 LVM 而不是 udev 规则来管理 LV 设备节点和符号链接，可以使用下列方法之一来禁止 udev 发出通知：

- 在 `/etc/lvm/lvm.conf` 中配置 `activation/udev_rules = 0` 和 `activation/udev_sync = 0`。

  请注意，结合 **lvcreate** 命令指定 `--nodevsync` 的效果与设置 `activation/udev_sync = 0` 相同；仍需设置 `activation/udev_rules = 0`。

- 设置环境变量 `DM_DISABLE_UDEV`：

  ```
  export DM_DISABLE_UDEV=1
  ```

  这样也会禁止 udev 发出通知。此外，`/etc/lvm/lvm.conf` 中所有 udev 相关设置将被忽略。

5.8.1 使用命令调整逻辑卷的大小

lvresize、**lvextend** 和 **lvreduce** 命令用于调整逻辑卷大小。有关语法和选项信息，请参见每个命令的手册页。要扩展 LV，VG 上必须有足够的可用未分配空间。

建议使用 YaST 分区程序来增大或缩小逻辑卷。使用 YaST 时，卷中文件系统的大小也会相应进行自动调整。

LV 可以在使用中状态下进行手动增大或缩小，但是其上的文件系统无法随之调整。扩展或收缩 LV 不会自动修改卷中文件系统的大小。随后必须使用其他命令增大文件系统。有关调整文件系统大小的信息，请参见第 2 章 “调整文件系统的大小”。

手动调整 LV 大小时，请确保使用正确的顺序：

- 如果扩展逻辑卷，则必须在试图增大文件系统之前扩展逻辑卷。
- 如果缩小逻辑卷，则必须在试图缩小逻辑卷之前缩小文件系统。

扩展逻辑卷的大小：

1. 打开一个终端控制台。
2. 如果逻辑卷包含 Ext2 或 Ext4 文件系统，则不支持联机增大，请将其卸下。如果它包含为虚拟机（例如 Xen VM）托管的文件系统，请先关闭该 VM。
3. 在终端控制台提示时，输入以下命令以增加逻辑卷大小：

   ```
   > sudo lvextend -L +SIZE /dev/VG_NAME/LV_NAME
   ```

   对于 SIZE，请指定要添加到逻辑卷的空间容量，例如 10 GB。将 /dev/VG_NAME/LV_NAME 替换为逻辑卷的 Linux 路径，例如 /dev/LOCAL/DATA。例如：

   ```
   > sudo lvextend -L +10GB /dev/vg1/v1
   ```

4. 调整文件系统的大小。有关详细信息，请参见第 2 章 “调整文件系统的大小”。
5. 如果卸下了文件系统，请将其重新挂载。

例如，将带有（已挂载并激活）Btrfs 的 LV 扩大 10GB：

```
> sudo lvextend –L +10G /dev/LOCAL/DATA
```

```
> sudo btrfs filesystem resize +10G /dev/LOCAL/DATA
```

要缩小逻辑卷的大小，请执行以下操作：

1. 打开一个终端控制台。
2. 如果逻辑卷不包含 Btrfs 文件系统，请将其卸下。如果它包含为虚拟机（例如 Xen VM）托管的文件系统，请先关闭该 VM。请注意，带有 XFS 文件系统的卷不能减少大小。
3. 调整文件系统的大小。有关详细信息，请参见 第 2 章 “调整文件系统的大小”。
4. 在终端控制台提示符中，输入下列命令以将逻辑卷的大小缩小为文件系统的大小：

   ```
   > sudo lvreduce /dev/VG_NAME/LV_NAME
   ```

5. 如果卸载了文件系统，请将其重新挂载。

例如，将带有 Btrfs 的 LV 缩减 5GB：

```
> sudo btrfs filesystem resize -size 5G /dev/LOCAL/DATA
sudo lvreduce /dev/LOCAL/DATA
```

提示： 使用一条命令调整卷和文件系统的大小

从 SUSE Linux Enterprise Server 12 SP1 开始，**`lvextend`**、**`lvresize`** 和 **`lvreduce`** 支持 `--resizefs` 选项，该选项不仅可以更改卷的大小，而且还能调整文件系统的大小。因此，上面所示的 **`lvextend`** 和 **`lvreduce`** 示例也可以改编为：

```
> sudo lvextend --resizefs –L +10G /dev/LOCAL/DATA
> sudo lvreduce  --resizefs -L -5G /dev/LOCAL/DATA
```

请注意，以下文件系统支持 `--resizefs`：ext2/3/4、Btrfs 和 XFS。目前只能在 SUSE Linux Enterprise Server 上使用此选项调整 Btrfs 的大小，因为上游尚未接受此选项。

5.8.2 使用 LVM 缓存卷

LVM 支持使用高速块设备（例如 SSD 设备）作为较低速大型块设备的写回或直写缓存。该缓存逻辑卷类型使用小型高速 LV 来提高大型慢速 LV 的性能。

要设置 LVM 缓存，需在缓存设备上创建两个逻辑卷。较大的卷用于缓存本身，较小的卷用于储存缓存元数据。这两个卷需属于原始卷所在的同一个卷组。创建这些卷后，需要将其转换为缓存池，并将该池挂接到原始卷：

过程 5.2： **设置已缓存的逻辑卷**

1. 在慢速设备上创建原始卷（如果尚不存在）。
2. 将物理卷（从快速设备）添加到原始卷所属的同一个卷组，然后在物理卷上创建缓存数据卷。
3. 创建缓存元数据卷。该卷的大小应为缓存数据卷大小的 1/1000，最小大小为 8 MB。
4. 将缓存数据卷和元数据卷组合成一个缓存池卷：

```
> sudo lvconvert --type cache-pool --poolmetadata VOLUME_GROUP/
METADATA_VOLUME VOLUME_GROUP/CACHING_VOLUME
```

5. 将缓存池挂接到原始卷：

```
> sudo lvconvert --type cache --cachepool VOLUME_GROUP/
CACHING_VOLUME VOLUME_GROUP/ORIGINAL_VOLUME
```

有关 LVM 缓存的详细信息，请参见 lvmcache(7) 手册页。

5.9 标记 LVM2 储存对象

标记是无序的关键字或指派给储存对象元数据的术语。使用标记功能可以采用您认为有用的方式将无序的标记列表附加到 LVM 储存对象元数据，从而对储存对象集合进行分类。

5.9.1 使用 LVM2 标记

标记 LVM2 储存对象后，可以在命令中使用标记来完成以下任务：

- 根据有或无特定标记选择要处理的 LVM 对象。
- 使用配置文件中的标记，控制在服务器上激活哪些卷组和逻辑卷。
- 通过在命令中指定标记，覆盖全局配置中的设置。

可以使用标记代替接受以下项的任何命令行 LVM 对象参考：

- 对象列表
- 单个对象，只要标记扩展到单个对象

目前，在任何位置都不支持将对象名称替换为标记。扩展参数后，列表中的重复参数将通过删除重复参数而保留每个参数的第一个实例来解决。

只要是参数类型可能不明确的位置，都必须在标记前面加上商业性 (@) 字符，比如 `@mytag`。在其他位置是否使用 “@” 前缀是可选的。

5.9.2 创建 LVM2 标记的要求

对 LVM 使用标记时，请注意以下要求：

支持的字符

LVM 标记单词可以包含 ASCII 大写字符 A 到 Z、小写字符 a 到 z、数字 0 到 9、下划线 (_)、加号 (+)、连字符 (-) 和句点 (.)。单词不能以连字符开头。最大长度为 128 个字符。

支持的储存对象

可以标记 LVM2 物理卷、卷组、逻辑卷和逻辑卷分段。PV 标记将储存在其卷组的元数据中。删除卷组时也删除孤立的物理卷中的标记。快照不能标记，但快照源可以标记。LVM1 对象不能标记，因为磁盘格式不支持此功能。

5.9.3 命令行标记语法

`--addtagTAG_INFO`

将标记添加到（或**标记**）LVM2 储存对象。示例：

```
> sudo vgchange --addtag @db1 vg1
```

`--deltagTAG_INFO`

删除（或**取消标记**）LVM2 储存对象中的标记。示例：

```
> sudo vgchange --deltag @db1 vg1
```

`--tagTAG_INFO`

指定用于缩小要激活或停用的卷组列表或逻辑卷的标记。

如果卷的标记与提供的标记匹配，则输入以下命令将其激活（示例）：

```
> sudo lvchange -ay --tag @db1 vg1/vol2
```

5.9.4 配置文件语法

以下几节显示了特定用例的示例配置。

5.9.4.1 在 lvm.conf 文件中启用主机名标记

将下面的代码添加到 `/etc/lvm/lvm.conf` 文件中，以启用主机上 `/etc/lvm/lvm_<HOSTNAME>.conf` 文件中单独定义的主机标记。

```
tags {
   # Enable hostname tags
   hosttags = 1
}
```

您将激活代码置于主机上的 `/etc/lvm/lvm_<HOSTNAME>.conf` 文件中。请参见第 5.9.4.3 节“定义激活”。

5.9.4.2 在 lvm.conf 文件中为主机名定义标记

```
tags {

   tag1 { }
      # Tag does not require a match to be set.

   tag2 {
      # If no exact match, tag is not set.
      host_list = [ "hostname1", "hostname2" ]
   }
```

```
}
```

5.9.4.3 定义激活

您可以修改 /etc/lvm/lvm.conf 文件，以根据标记激活 LVM 逻辑卷。

在文本编辑器中，将以下代码添加到文件中：

```
activation {
    volume_list = [ "vg1/lvol0", "@database" ]
}
```

用您的标记替换 @database。使用 "@*" 根据主机上设置的任何标记匹配此标记。

激活命令会匹配卷组和逻辑卷的元数据中设置的 VGNAME、VGNAME/LVNAME 或 @TAG。只有元数据标记匹配时，才激活卷组或逻辑卷。默认情况下，如果不匹配则不会激活。

如果 volume_list 不存在，且主机上已定义标记，则只有主机标记与元数据标记匹配时，才会激活卷组或逻辑卷。

如果 volume_list 已定义但为空，并且主机上未定义任何标记，则不会激活。

如果未定义 volume_list，则不会施加任何 LV 激活限制（允许所有情况）。

5.9.4.4 在多个主机名配置文件中定义激活

在 lvm.conf 文件中启用主机标记后，便可在主机配置文件 (/etc/lvm/lvm_<HOST_TAG>.conf) 中使用激活代码。例如，一个服务器在 /etc/lvm/ 目录中有两个配置文件：

lvm.conf

lvm_<HOST_TAG>.conf

启动时，会装载 /etc/lvm/lvm.conf 文件，并处理该文件中的任何标记设置。如果定义了任何主机标记，它会装载相关的 /etc/lvm/lvm_<HOST_TAG>.conf 文件。搜索特定配置文件项目时，将会先搜索主机标记文件，然后搜索 lvm.conf 文件，并在找到第一个匹配项时停止搜索。在 lvm_<HOST_TAG>.conf 文件中，使用与所设标记顺序相反的顺序。这样就可以先搜索最后设置标记的文件。主机标记文件中设置的新标记将会触发额外的配置文件装载。

5.9.5 将标记用于群集中的简单激活控制

您可以通过在 `/etc/lvm/lvm.conf` 文件中启用 `hostname_tags` 选项来设置简单主机名激活控制。将相同文件用于群集中的每一台计算机上，以便建立全局设置。

1. 在文本编辑器中，将以下代码添加到 `/etc/lvm/lvm.conf` 文件中：

```
tags {
   hostname_tags = 1
}
```

2. 将文件复制到群集中的所有主机上。
3. 从群集中的任何计算机上，将 `db1` 添加到将激活 `vg1/lvol2` 的计算机列表中：

```
> sudo lvchange --addtag @db1 vg1/lvol2
```

4. 在 `db1` 服务器上，输入以下内容以激活它：

```
> sudo lvchange -ay vg1/vol2
```

5.9.6 使用标记激活群集中的首选主机

本部分中的示例展示了完成以下任务的两种方法：

- 仅激活数据库主机 `db1` 和 `db2` 上的卷组 `vg1`。
- 仅激活文件服务器主机 `fs1` 上的卷组 `vg2`。
- 最初在文件服务器备份主机 `fsb1` 上不激活任何卷组，但对其进行准备以从文件服务器主机 `fs1` 接管卷组。

5.9.6.1 选项 1：在主机间复制的集中化管理和静态配置

在以下解决方案中，单个配置文件复制到多台主机中。

1. 将 `@database` 标记添加到卷组 `vg1` 的元数据中。在终端控制台上，输入

```
> sudo vgchange --addtag @database vg1
```

2. 将 `@fileserver` 标记添加到卷组 `vg2` 的元数据中。在终端控制台上，输入

```
> sudo vgchange --addtag @fileserver vg2
```

3. 在文本编辑器中，用以下代码修改 `/etc/lvm/lvm.conf` 文件，以定义 `@database`、`@fileserver` 和 `@fileserverbackup` 标记。

```
tags {
   database {
      host_list = [ "db1", "db2" ]
   }
   fileserver {
      host_list = [ "fs1" ]
   }
   fileserverbackup {
      host_list = [ "fsb1" ]
   }
}

activation {
   # Activate only if host has a tag that matches a metadata tag
   volume_list = [ "@*" ]
}
```

4. 将修改后的 `/etc/lvm/lvm.conf` 文件复制到四台主机中：`db1`、`db2`、`fs1` 和 `fsb1`。

5. 如果文件服务器主机出现故障，则在终端控制台的任何节点上输入以下命令即可在 `fsb1` 上启动 `vg2`：

```
> sudo vgchange --addtag @fileserverbackup vg2
> sudo vgchange -ay vg2
```

5.9.6.2 选项 2：本地化管理和配置

在以下解决方案中，每台主机在本地保存有关激活哪些类别的卷的信息。

1. 将 @database 标记添加到卷组 vg1 的元数据中。在终端控制台上，输入

```
> sudo vgchange --addtag @database vg1
```

2. 将 @fileserver 标记添加到卷组 vg2 的元数据中。在终端控制台上，输入

```
> sudo vgchange --addtag @fileserver vg2
```

3. 在 /etc/lvm/lvm.conf 文件中启用主机标记：

   a. 在文本编辑器中，用以下代码修改 /etc/lvm/lvm.conf 文件，以启用主机标记配置文件。

   ```
   tags {
      hosttags = 1
   }
   ```

   b. 将修改后的 /etc/lvm/lvm.conf 文件复制到四台主机上：db1、db2、fs1 和 fsb1。

4. 在主机 db1 上，创建数据库主机 db1 的激活配置文件。在文本编辑器中，创建 /etc/lvm/lvm_db1.conf 文件并添加以下代码：

```
activation {
   volume_list = [ "@database" ]
}
```

5. 在主机 db2 上，创建数据库主机 db2 的激活配置文件。在文本编辑器中，创建 /etc/lvm/lvm_db2.conf 文件并添加以下代码：

```
activation {
   volume_list = [ "@database" ]
}
```

6. 在主机 fs1 上，创建文件服务器主机 fs1 的激活配置文件。在文本编辑器中，创建 /etc/lvm/lvm_fs1.conf 文件并添加以下代码：

```
activation {
```

```
   volume_list = [ "@fileserver" ]
}
```

7. 如果文件服务器主机 `fs1` 出现故障，要启动备用文件服务器主机 fsb1 作为文件服务器：

   a. 在主机 `fsb1` 上，创建主机 `fsb1` 的激活配置文件。在文本编辑器中，创建 `/etc/lvm/lvm_fsb1.conf` 文件并添加以下代码：

   ```
   activation {
      volume_list = [ "@fileserver" ]
   }
   ```

   b. 在终端控制台上，输入以下命令之一：

   ```
   > sudo vgchange -ay vg2
   > sudo vgchange -ay @fileserver
   ```

6 LVM 卷快照

逻辑卷管理器 (LVM) 逻辑卷快照是一种写入时复制技术，它会监控现有卷数据块的更改，以便在对其中一个块执行写入操作时，将进行快照时块的值复制到快照卷。这样，便可保留数据的时间点副本，直到快照卷删除为止。

6.1 了解卷快照

文件系统快照包含有关自身的元数据以及在生成快照后更改过的源逻辑卷的数据块。通过快照访问数据时，您会看到复制来源逻辑卷的时间点。不需要从备份媒体恢复数据或重写更改过的数据。

重要：挂载含有快照的卷

在快照有效期内，必须先挂载快照，然后才能挂载其来源逻辑卷。

LVM 卷快照可用于从文件系统的时间点视图创建备份。快照是即时创建并永久保留的，直到您将其删除为止。您可以从快照备份文件系统，而卷本身仍可继续供用户使用。快照最初包含自身相关的一些元数据，但不包含来源逻辑卷的实际数据。快照使用写时复制技术在原始数据块中的数据发生更改时进行检测。当对快照卷中的块捕获快照时，它会复制所包含的值，然后允许在来源块中储存新的数据。随着来源逻辑卷上有更多块更改其原始值，快照大小将会增大。

调整快照大小时，请考虑来源逻辑卷中要更改的数据量，以及要保留快照的时间。您为快照卷分配的空间量因以下因素而异：来源逻辑卷的大小、计划保留快照的时间，以及在快照有效期中预期会更改的数据块数。快照卷创建后不能调整大小。从原则上说，应创建一个约占原始逻辑卷大小 10% 的快照卷。如果您预测在删除快照前，来源逻辑卷中的每个块都会至少更改一次，则快照的容量至少应相当于来源逻辑卷的容量加上部份额外空间，其中后者用于储存快照卷的相关元数据。如果数据更改不那么频繁或如果预期的有效期足够短，则需要的空间较少。

在 LVM2 中，快照默认为读/写。直接将数据写入快照时，该块在例外表格中标示为使用，不会从来源逻辑卷中复制。您可以装载快照卷，并通过直接将数据写入快照卷来测试应用更改。您可以通过卸载快照、去除快照，然后重新装载来源逻辑卷，轻松丢弃更改。

在虚拟 Guest 环境中，您可以使用快照功能用于在服务器的磁盘上创建的 LVM 逻辑卷，就如在物理服务器上一样。

在虚拟主机环境中，您可以使用快照功能来备份虚拟机的储存后端，或测试对虚拟机映像（例如用于修补程序或升级）进行的更改，而不必修改来源逻辑卷。虚拟机必须将 LVM 逻辑卷用做其储存后端，以免使用虚拟磁盘文件。您可以装载 LVM 逻辑卷，并将它用做文件型磁盘来储存虚拟机映像；也可以指派 LVM 逻辑卷做为物理磁盘，以便将其视为块设备进行写入操作。

从 SLES 11 SP3 开始，LVM 逻辑卷快照可以瘦配置。如果您创建没有指定大小的快照，则会使用瘦配置。创建为瘦卷的快照在需要时使用瘦池中的空间。快照瘦卷的特性与任何其他瘦卷相同。您可以独立地激活卷、扩展卷、重命名卷、去除卷，甚至可以创建卷的快照。

重要：群集中精简配置的卷

若要使用群集中瘦配置的快照，来源逻辑卷及其快照必须在单个群集资源中管理。这允许卷及其快照在同一个节点上始终独占性地装载。

当用完快照后，一定要将其从系统中删除。随着来源逻辑卷上数据块的不断更改，快照终将完全填满。填满时就会处于禁用状态，导致您无法重新装载来源逻辑卷。

如果您为一个来源逻辑卷创建多个快照，在去除最后创建快照之前，请先删除较旧的快照。

6.2 使用 LVM 创建 Linux 快照

可以使用逻辑卷管理器 (LVM) 创建文件系统的快照。

打开终端控制台，然后输入

```
> sudo lvcreate -s [-L <size>] -n SNAP_VOLUME SOURCE_VOLUME_PATH
```

如果不指定大小，快照会创建为瘦快照。

例如：

```
> sudo lvcreate -s -L 1G -n linux01-snap /dev/lvm/linux01
```

快照将被创建为 `/dev/lvm/linux01-snap` 卷。

6.3 监视快照

打开终端控制台，然后输入

```
> sudo lvdisplay SNAP_VOLUME
```

例如：

```
> sudo lvdisplay /dev/vg01/linux01-snap

--- Logical volume ---
  LV Name                /dev/lvm/linux01
  VG Name                vg01
  LV UUID                QHVJYh-PR3s-A4SG-s4Aa-MyWN-Ra7a-HL47KL
  LV Write Access        read/write
  LV snapshot status     active destination for /dev/lvm/linux01
  LV Status              available
  # open                 0
  LV Size                80.00 GB
  Current LE             1024
  COW-table size         8.00 GB
  COW-table LE           512
  Allocated to snapshot  30%
  Snapshot chunk size    8.00 KB
  Segments               1
  Allocation             inherit
  Read ahead sectors     0
  Block device           254:5
```

6.4 删除 Linux 快照

打开终端控制台，然后输入

```
> sudo lvremove SNAP_VOLUME_PATH
```

例如：

```
> sudo lvremove /dev/lvmvg/linux01-snap
```

6.5 在虚拟主机上使用虚拟机的快照

如果将 LVM 逻辑卷用做虚拟机的后端储存，可以让系统灵活地管理基础设备，例如更轻松地移动储存对象、创建快照和备份数据。您可以装载 LVM 逻辑卷，并将它用做文件型磁盘来储存虚拟机映像；也可以指派 LVM 逻辑卷做为物理磁盘，以便将其视为块设备进行写入操作。您可以在 LVM 逻辑卷上创建虚拟磁盘映像，然后创建 LVM 快照。

您可以利用快照的读/写功能创建虚拟机的不同实例，并在这些实例中更改特定虚拟机实例的快照。您也可以在 LVM 逻辑卷上创建虚拟磁盘映像、创建来源逻辑卷的快照以及修改特定虚拟机实例的快照。您还可以创建来源逻辑卷的另一个快照，并修改该快照以取得不同虚拟机实例。不同虚拟机实例的大部分数据与映像一起储存在来源逻辑卷上。

在 Guest 环境中，您还可以利用快照的读/写功能保留虚拟磁盘映像，同时测试修补程序或升级。您创建包含映像的 LVM 卷的快照，然后在快照位置运行虚拟机。来源逻辑卷保持不变，对机器的所有更改均写入快照。为了恢复到虚拟机映像的来源逻辑卷，您需要关闭虚拟机，然后从来源逻辑卷中去除快照。若要重新开始，请重新创建快照、装载快照，然后在快照映像上重启动虚拟机。

下列程序使用文件型虚拟磁盘映像和 Xen 超级管理程序。对于在 SUSE Linux Enterprise 平台上运行的其他超级管理程序（例如 KVM），您可以调整本节中的过程。若要从快照卷中运行文件型虚拟机映像，请运行下列步骤：

1. 确保已挂载包含文件型虚拟机映像的来源逻辑卷，例如在挂载点 `/var/lib/xen/images/<IMAGE_NAME>` 挂载。

2. 创建具有足够空间来储存预期差别的 LVM 逻辑卷快照。

   ```
   > sudo lvcreate -s -L 20G -n myvm-snap /dev/lvmvg/myvm
   ```

   如果不指定大小，快照会创建为瘦快照。

3. 创建挂载点，用于挂载快照卷。

   ```
   > sudo mkdir -p /mnt/xen/vm/myvm-snap
   ```

4. 在所创建的挂载点挂载快照卷。

   ```
   > sudo mount -t auto /dev/lvmvg/myvm-snap /mnt/xen/vm/myvm-snap
   ```

5. 在文本编辑器中，复制来源虚拟机的配置文件，修改指向挂载快照卷上的文件型图像文件的路径，然后储存文件，例如 `/etc/xen/myvm-snap.cfg`。

6. 使用虚拟机的已挂载快照卷启动虚拟机。

```
> sudo xm create -c /etc/xen/myvm-snap.cfg
```

7. （可选）去除快照，然后在来源逻辑卷使用未更改的虚拟机映像。

```
> sudo umount /mnt/xenvms/myvm-snap
> sudo lvremove -f /dev/lvmvg/mylvm-snap
```

8. （可选）根据需要重复此程序。

6.6 将快照与来源逻辑卷合并以还原更改或回滚到先前的状态

如果您需要将卷上的数据回滚或还原至先前的状态，快照可能非常有用。例如，如果因管理员失误，或是因软件包安装或升级出故障或没必要，导致数据出现更改，您可能需要予以还原。

您可以使用 **`lvconvert --merge`** 命令还原对 LVM 逻辑卷的更改。合并按如下所示开始：

- 如果来源逻辑卷和快照卷均未打开，合并将立即开始。
- 如果来源逻辑卷或快照卷已打开，合并将在来源逻辑卷或快照卷第一次启动和同时关闭时开始。
- 如果来源逻辑卷不能关闭（例如根文件系统），系统会推迟到下一次服务器重引导并激活来源逻辑卷的时候再合并。
- 如果来源逻辑卷包含虚拟机映像，您必须关闭虚拟机，停用来源逻辑卷和快照卷（也就是依次卸下这些卷），然后发出合并命令。因为来源逻辑卷会自动重新挂载，而且合并完成时会删除快照卷，所以在合并完成之前请勿重启动虚拟机。合并完成之后，您可以将生成的逻辑卷用于虚拟机。

合并开始之后，系统将在服务器重启动之后继续合并，直到合并完成。在合并期间，无法为来源逻辑卷创建新快照。

合并期间，系统对来源逻辑卷的读取或写入操作会透明地重定向到正在合并的快照，因此用户能够立即查看和访问数据，就像当时创建快照时一样，不必等到合并完成。

合并完成后，来源逻辑卷会包含与创建快照时相同的数据，加上合并开始后对数据的任何更改。生成的逻辑卷沿用了来源逻辑卷的名称、次要编号和 UUID。系统会自动重新挂载来源逻辑卷，并去除快照卷。

1. 打开终端控制台，然后输入

```
> sudo lvconvert --merge  [-b] [-i SECONDS] [SNAP_VOLUME_PATH[...snapN]|
@VOLUME_TAG]
```

您可以在命令行上指定一或多个快照；也可以使用相同卷标记来标记多个来源逻辑卷，然后在命令行上指定 @<VOLUME_TAG>。标记过的卷的快照会合并到其各自的来源逻辑卷中。有关标记逻辑卷的的相关信息，请参见第 5.9 节 “标记 LVM2 储存对象”。
选项包括：

-b,
--background
在背景中执行守护程序，这样可以并行合并多个指定的快照。

-i,
--interval <SECONDS>
以固定的间隔以百分比形式报告进度。指定的间隔以秒为单位。

有关此命令的详细信息，请参见 **lvconvert(8)** man 页面。
例如：

```
> sudo lvconvert --merge /dev/lvmvg/linux01-snap
```

此命令将 /dev/lvmvg/linux01-snap 合并到其来源逻辑卷中。

```
> sudo lvconvert --merge @mytag
```

如果 lvol1、lvol2 和 lvol3 全都以 mytag 标记，每个快照卷将依序与其各自的来源逻辑卷合并；即先 lvol1，再 lvol2，然后 lvol3。如果指定了 --background 选项，相应标记过的逻辑卷的快照将并行同时合并。

2. （可选）如果来源逻辑卷和快照卷均已打开且可以关闭，则您可以手动停用然后激活来源逻辑卷，以便让合并立即开始。

```
> sudo umount ORIGINAL_VOLUME
> sudo lvchange -an ORIGINAL_VOLUME
> sudo lvchange -ay ORIGINAL_VOLUME
> sudo mount ORIGINAL_VOLUME MOUNT_POINT
```

例如：

```
> sudo umount /dev/lvmvg/lvol01
> sudo lvchange -an /dev/lvmvg/lvol01
> sudo lvchange -ay /dev/lvmvg/lvol01
> sudo mount /dev/lvmvg/lvol01 /mnt/lvol01
```

3. （可选）如果来源逻辑卷和快照卷均已打开且来源逻辑卷不能关闭（例如 `root` 文件系统），则您可以重启动服务器并挂载来源逻辑卷，以让合并在重启动之后立即开始。

III 软件 RAID

7 软件 RAID 配置 90

8 为根分区配置软件 RAID 97

9 创建软件 RAID 10 设备 104

10 创建降级 RAID 阵列 118

11 使用 mdadm 调整软件 RAID 阵列的大小 120

12 适用于 MD 软件 RAID 的储存机箱 LED 实用程序 128

13 软件 RAID 查错 136

7 软件 RAID 配置

RAID（独立磁盘冗余阵列）的用途是将多个硬盘分区合并成一个大的虚拟硬盘，以便优化性能和/或数据安全性。大多数 RAID 控制器使用 SCSI 协议，因为与 IDE 协议相比，它可使用更高效的方式处理更大量的硬盘，并且更适合命令的并行处理。还有一些支持 IDE 或 SATA 硬盘的 RAID 控制器。软件 RAID 具有 RAID 系统的优势，同时没有硬件 RAID 控制器的额外成本。但是这需要一些 CPU 时间以及内存，所以不适用于真正高性能的计算机。

重要：群集文件系统上的 RAID

需要使用群集多设备（群集 MD）来设置群集文件系统下的软件 RAID。请参见《SUSE Linux Enterprise High Availability Extension 管理指南》(https://documentation.suse.com/sle-ha/15-SP2/html/SLE-HA-all/cha-ha-cluster-md.html)。

SUSE Linux Enterprise 提供了将若干硬盘组合为一个软 RAID 系统的选项。RAID 暗示将多块硬盘合成一个 RAID 系统的多种策略，这些策略的目标、优点及特点各不相同。这些变化形式通常称作 **RAID 级别**。

7.1 了解 RAID 级别

本节描述常见的 RAID 级别 0、1、2、3、4、5 和嵌套的 RAID 级别。

7.1.1 RAID 0

此级别通过将每个文件按块分放到多个磁盘上，提高了数据访问性能。此级别实际上并不是 RAID，因为它不提供数据备份，但 **RAID 0** 已成为这种系统类型的标准名称。使用 RAID 0，可以将两块或多块硬盘组合在一起。这样性能固然很好，但如果有任何一块硬盘出现故障，都将损坏 RAID 系统并丢失数据。

7.1.2 RAID 1

此级别为您的数据提供了足够的安全性，因为数据会 1:1 复制到另一个硬盘。这称为**硬盘镜像**。如果一块磁盘损坏，则可以使用另一块镜像磁盘上的内容副本。在所有这些硬盘中，只要有一块硬盘没有损坏，您的数据就不会丢失。但是，如果没有检测到损坏，损坏的数据可能会镜像到正确的磁盘，并以这种方式损坏其数据。与使用单个磁盘访问时相比，写性能在复制过程中稍有损失（慢 10% 到 20%），但读访问的速度要大大快于任何一块普通物理硬盘，原因是对数据进行了复制，从而可以并行扫描它们。RAID 1 通常提供几乎为单个磁盘读事务速率两倍的速率，写事务速率与单个磁盘几乎相同。

7.1.3 RAID 2 和 RAID 3

这些不是典型的 RAID 实现。级别 2 在位一级而不是块一级对数据进行分段。级别 3 则利用专用的校验磁盘在字节一级进行分段，但不能同时处理多个请求。这两种级别都极少使用。

7.1.4 RAID 4

级别 4 与级别 0 一样，也是在块一级进行分段，但结合使用了专用的校验磁盘。如果一块数据磁盘失败，将使用奇偶校验数据创建替换磁盘。不过，这块奇偶校验磁盘可能造成写访问的瓶颈。尽管如此，有时仍使用级别 4。

7.1.5 RAID 5

RAID 5 是级别 0 和级别 1 在性能和冗余方面经优化后的折衷方案。硬盘空间等于使用的磁盘数减 1。数据使用与 RAID 0 相同的方式分布到硬盘中。在其中一个分区上创建的**奇偶校验块**是基于安全考虑。这些块通过 XOR 互相链接，并在系统出现故障时，通过启用相应的校验块重构建内容。对于 RAID 5，在同一时间只能有一块硬盘出现故障。如果一块硬盘出现故障，则必须在情况允许时将其更换，以防止丢失数据。

7.1.6 RAID 6

RAID 6 是 RAID 5 的扩展，它通过使用第二种独立分布式奇偶校验模式（双重奇偶校验）来增加容错能力。即使在数据恢复过程中两个硬盘出现故障，系统仍将继续运行，数据不会丢失。

RAID 6 可承受多个并行驱动器故障，从而提供非常高的数据容错性能。它处理任何两个设备的丢失而不会丢失数据。此外，它还需要 N+2 个驱动器来储存相当于 N 个驱动器的数据。它至少需要四个设备。

RAID 6 的性能稍微低一些，但在正常模式和单磁盘故障模式下可以与 RAID 5 媲美。它在双磁盘故障模式下非常慢。RAID 6 配置需要占用相当多的 CPU 时间和内存，用于写入操作。

表 7.1： RAID 5 和 RAID 6 的比较

功能	RAID 5	RAID 6
设备数	N+1，至少 3 个	N+2，至少 4 个
奇偶校验	分布式，单	分布式，双
性能	在写和重构建方面有中度影响	比 RAID 5 在串行写方面影响大
容错	一个组件设备的故障	两个组件设备的故障

7.1.7 嵌套和复杂 RAID 级别

现在已开发出了其他 RAID 级别，例如 RAIDn、RAID 10、RAID 0+1、RAID 30 和 RAID 50。有些是硬件供应商创建的专用实施。创建 RAID 10 配置的示例可在第 9 章 “创建软件 RAID 10 设备”中找到。

7.2 使用 YaST 配置软件 RAID

可以通过 YaST 专家分区程序访问 YaST 软 RAID 配置。此分区工具还用于编辑和删除现有分区，并创建用于软 RAID 的新分区。下列说明在设置 RAID 级别 0、1、5 和 6 时适用。设置 RAID 10 配置的方法如第 9 章 “创建软件 RAID 10 设备”所述。

1. 启动 YaST 并打开分区程序。

2. 如果需要，请创建应该与 RAID 配置搭配使用的分区。请勿将它们格式化，并将分区类型设置为 0xFD Linux RAID。使用现有分区时，不需要更改它们的分区类型 — YaST 会自动更改。有关细节，请参考《部署指南》, 第 10 章 “专家分区程序”, 第 10.1 节 “使用专家分区程序”。

   强烈建议您使用储存在其他硬盘上的分区，以便降低当其中一个硬盘损坏时（RAID 1 和 5）遗失数据的风险，并优化 RAID 0 的性能。

   对于 RAID 0，至少需要两个分区。RAID 1 只需要两个分区，而 RAID 5 至少需要三个分区。RAID 6 设置至少需要四个分区。建议仅使用大小相同的分区，因为每个段仅可将相同的空间量作为最小大小的分区。

3. 在左侧面板中，选择 RAID。

   右侧面板中即会打开现有 RAID 配置的列表。

4. 在 RAID 页面的左下方，单击 添加 RAID。

5. 选择RAID 类型并从可用设备对话框中添加适当数目的分区。

   您可以选择性地为 RAID 指派一个 RAID 名称。这样，RAID 的名称将是 `/dev/md/NAME`。有关更多信息，请参见第 7.2.1 节 “RAID 名称”。

图 7.1： RAID 5 配置示例

单击下一步继续。

6. 选择大块大小，如果适用，同时选择奇偶校验算法。最佳的大块大小视数据的类型和 RAID 的类型而定。有关更多信息，请参见https://raid.wiki.kernel.org/index.php/RAID_setup#Chunk_sizes。有关奇偶校验算法的详细信息，请在搜索 `--layout` 选项时加上 **`man 8 mdadm`**。如果不确定，请接受默认值。

7. 选择卷的角色。您在此处所做选择只会影响将要打开的对话框的默认值。这些值可在下一个步骤中更改。如果不确定，请选择原始卷（未格式化）。

8. 在格式化选项下，选择格式化分区，然后选择文件系统。选项菜单的内容取决于文件系统。通常不需要更改默认值。
   在挂载选项下，选择挂载分区，然后选择挂载点。单击 Fstab 选项，为卷添加特殊挂载选项。

9. 单击完成。

10. 单击下一步，校验是否列出了更改，然后单击完成。

重要：磁盘上的 RAID

虽然使用分区程序可以在磁盘（而不是分区）的顶层创建 RAID，但出于多种原因，我们不建议使用此方法。不支持在此类 RAID 上安装引导加载程序，因此您需要使用单独的设备进行引导。`fdisk` 和 `parted` 等工具在此类 RAID 上无法正常工作，不清楚 RAID 特定设置的人员在使用这些工具时，可能会做出错误的诊断或执行错误的操作。

7.2.1 RAID 名称

软件 RAID 设备默认带有 `mdN` 模式的数值名称，其中 `N` 是数字。因此，以 md127 为例，它可以按 `/dev/md127` 这样的方式进行访问，并且在 `/proc/mdstat` 和 `/proc/partitions` 中会列出为 `md127`。但这些名称不方便使用，为此，SUSE Linux Enterprise Server 提供了两种解决方法：

提供指向设备的具名链接

当您使用 YaST 或在命令行上使用 **mdadm --create '/dev/md/** NAME' 创建 RAID 设备时，可以选择为 RAID 设备指定一个名称。设备名称仍然是 mdN，但系统会创建一个链接 /dev/md/NAME：

```
> ls -og /dev/md
total 0
lrwxrwxrwx 1 8 Dec  9 15:11 myRAID -> ../md127
```

设备在 /proc 下仍显示为 md127。

提供具名设备

如果指向设备的具名链接不足以满足您的设置要求，请运行下列命令将 CREATE names=yes 行添加至 /etc/mdadm.conf：

```
> sudo echo "CREATE names=yes" >> /etc/mdadm.conf
```

此操作会让系统将 myRAID 之类的名称用做“真实”的设备名称。设备不但可以在 /dev/myRAID 中访问，而且也会在 /proc 下列为 myRAID。请注意，只有在更改配置文件之后，此项才适用于 RAID。活动 RAID 将继续使用 mdN 名称，直到这些 RAID 停止并重新组装为止。

警告：不兼容的工具

不是所有工具都可以支持具名 RAID 设备。如果工具认为一个 RAID 设备是命名为 mdN，它将无法识别该设备。

7.3 在 AArch64 上的 RAID 5 中配置条带大小

默认情况下，条带大小设置为 4kB。如果您需要更改默认条带大小（例如，使该大小与 AArch64 上的典型页面大小 64kB 相匹配），可以使用 CLI 手动配置条带大小：

```
> sudo echo 16384  > /sys/block/md1/md/stripe_size
```

以上命令将条带大小设置为 16kB。您可以设置其他值（例如 4096、8192），但该值必须是 2 的乘方。

7.4 监视软件 RAID

您可以在 `monitor` 模式下作为守护程序运行 **`mdadm`** 来监视软件 RAID。在 `monitor` 模式下，**`mdadm`** 会定期检查阵列中的磁盘故障。如果发生故障，**`mdadm`** 会向管理员发送一封电子邮件。要定义检查的时间间隔，请运行以下命令：

```
mdadm --monitor --mail=root@localhost --delay=1800 /dev/md2
```

上面的命令会每隔 1800 秒开启一次 `/dev/md2` 阵列监视。如果发生故障，将向 `root@localhost` 发送一封电子邮件。

注意：默认会启用 RAID 检查

默认情况下，RAID 检查处于启用状态。如果每次检查之间的间隔时间不够长，您可能会遇到警告。因此，您可以使用 `delay` 选项设置更大的值来增加时间间隔。

7.5 更多信息

位于下列位置的 HOWTO 文档提供了软 RAID 的配置说明和详细信息：

- Linux RAID wiki：https://raid.wiki.kernel.org/
- `/usr/share/doc/packages/mdadm/Software-RAID.HOWTO.html` 文件中的软件 RAID HOWTO

此外还提供了 Linux RAID 邮件列表，例如 linux-raid (http://marc.info/?l=linux-raid)。

8 为根分区配置软件 RAID

在 SUSE Linux Enterprise Server 中，设备映射程序 RAID 工具已集成到 YaST 分区程序中。安装时可以使用此分区程序为包含根 (/) 分区的系统设备创建软件 RAID。不能将 /boot 分区储存在除 RAID 1 以外的 RAID 分区上。

8.1 针对根分区使用软件 RAID 设备的先决条件

请确保配置满足以下要求：

- 创建 RAID 1 镜像设备需要有两块硬盘。这两块硬盘的大小应相近。RAID 使用较小驱动器大小。块储存设备可以是本地设备（计算机中或直接挂接到计算机）、光纤通道储存子系统或 iSCSI 储存子系统的任意组合。
- 如果在 MBR 中安装引导加载程序，则不需要为 /boot 创建独立分区。如果无法在 MBR 中安装引导加载程序，则 /boot 需要驻留在单独的分区上。
- 对于 UEFI 计算机，需要设置专用的 /boot/efi 分区。该分区需要格式化为 VFAT，可以驻留在 RAID 1 设备上，以防包含 /boot/efi 的物理磁盘发生故障时出现引导问题。
- 如果要使用硬件 RAID 设备，请不要在其上尝试运行软件 RAID。
- 如果要使用 iSCSI 目标设备，需要在创建 RAID 设备之前启用对 iSCSI 发起端的支持。
- 如果储存子系统在服务器和其直接挂接的本地设备、光纤通道设备或要在软件 RAID 中使用的 iSCSI 设备之间提供多个 I/O 路径，则在创建 RAID 设备之前需要启用多路径支持。

8.2 为根 (/) 分区设置具有软件 RAID 设备的系统

1. 使用 YaST 启动安装，并按照《部署指南》, 第 8 章 “安装步骤” 中的说明继续操作，直到达到建议的分区步骤为止。
2. 单击专家分区程序以打开自定义分区程序。您可以使用建议的方案，也可以使用现有方案。

3. （可选）如果有要使用的 iSCSI 目标设备，则需要从屏幕左上方的部分中选择系统 › 配置 › 配置 iSCSI，启用 iSCSI 发起端软件。有关更多细节，请参见第 15 章 “经由 IP 网络的大容量储存：iSCSI”。

4. （可选）如果有要使用的 FCoE 目标设备，则需要单击屏幕左上方的部分中的系统 › 配置 › 配置 FCoE 来配置界面。

5. （可选）如果您需要丢弃分区更改，请单击系统 › 重新扫描设备。

6. 为要用于软件 RAID 的每个设备设置 Linux RAID 格式。应该为 `/`、`/boot/efi` 或交换分区使用 RAID。

   a. 在左侧面板中，选择硬盘 并选择要使用的设备，然后单击添加分区。

   b. 在新分区大小下，指定要使用的大小，然后单击下一步。

   c. 在角色下，选择原始磁盘（未格式化）。

   d. 选择不格式化和不挂载，并将分区 ID 设置为 Linux RAID。

   e. 单击下一步，并对第二个分区重复这些步骤。

7. 为 `/` 分区创建 RAID 设备。

   a. 在左侧面板中，选择RAID，然后选择添加 RAID。

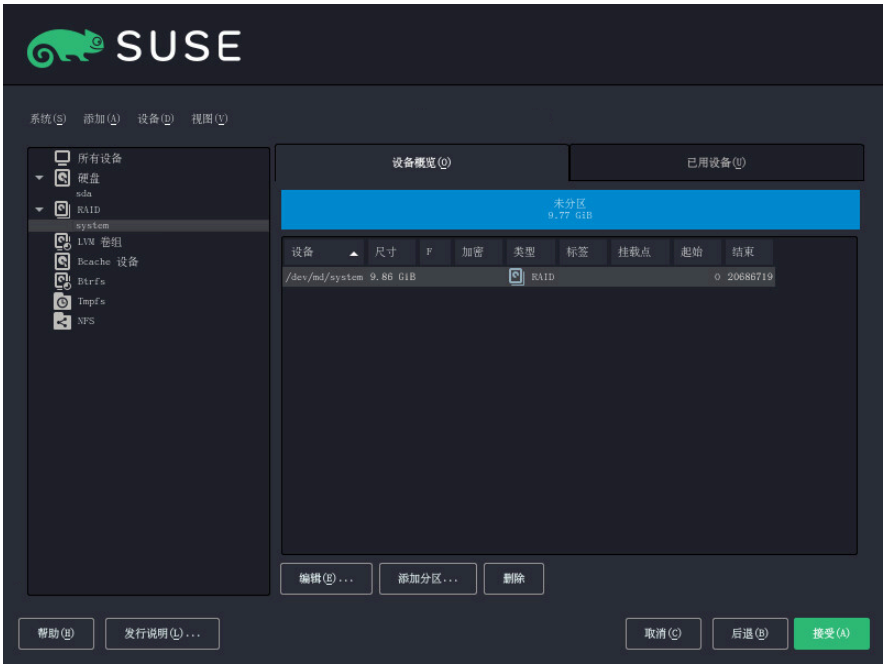
b. 为 / 分区设置所需的 RAID 类型，并将 RAID 名称设置为 `system`。

c. 从可用的设备部分中选择您在上一步中准备好的两个 RAID 设备并添加它们。

单击下一步继续。

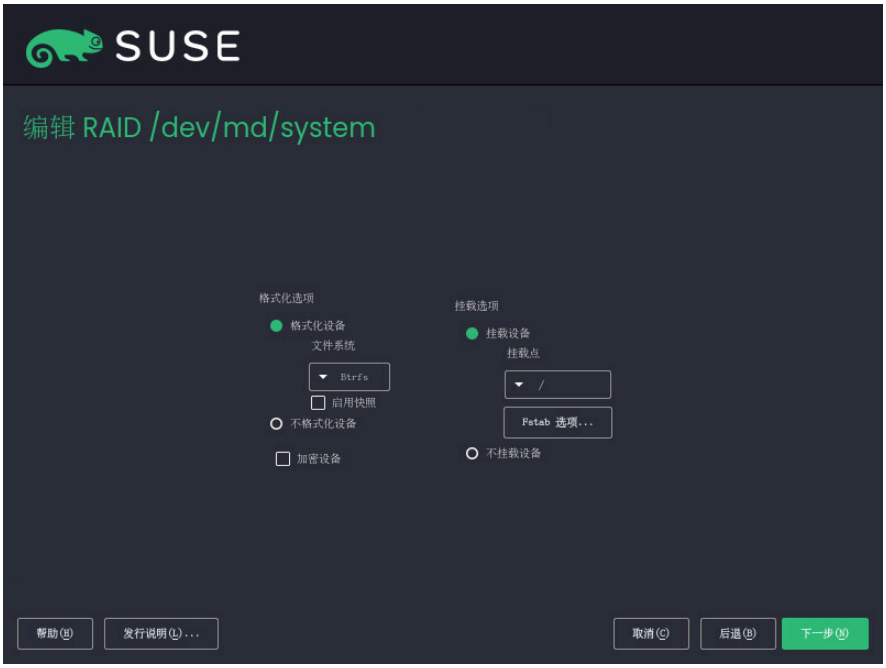
d. 从下拉框中选择区块大小。保留默认设置是安全的做法。

e. 在左侧面板中，单击 RAID。在设备概览选项卡中，选择新 RAID 并单击编辑。



f. 在角色下选择操作系统，然后单击下一步继续。

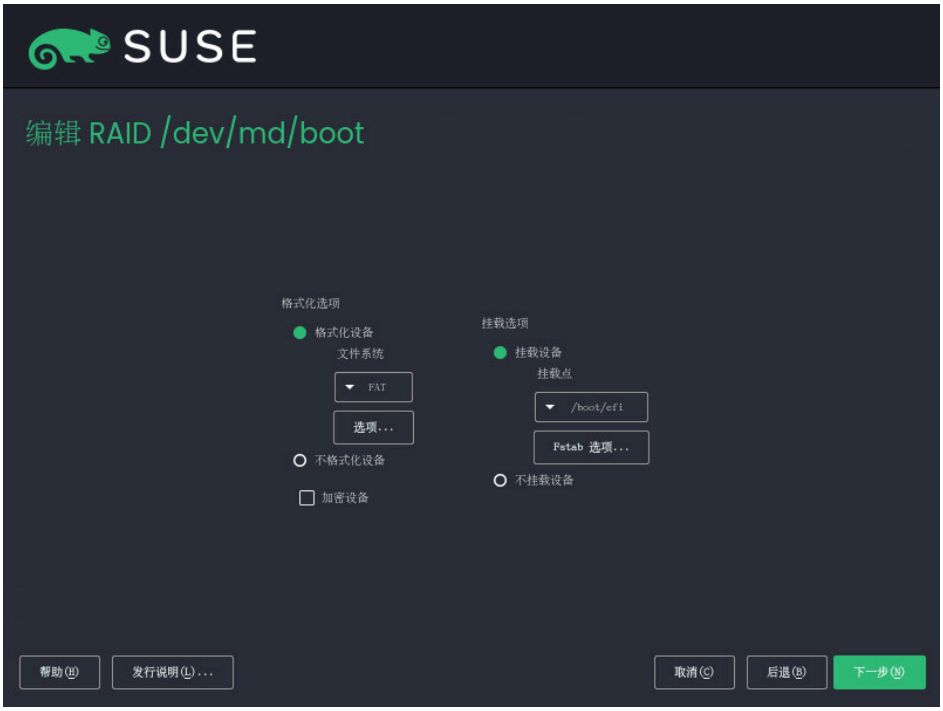
g. 选择文件系统并将挂载点设置为 /。单击下一步退出该对话框。



8. 软件 RAID 设备便会受设备映射程序的管理，并会在 /dev/md/system 路径下创建一个设备。

9. （可选）您可以在 RAID 中创建交换分区。使用与上述类似的步骤，只不过要在角色下选择交换。选择文件系统和挂载点，如下所示。单击下一步。

10. （可选）对于 UEFI 计算机，使用类似的步骤来创建 `/boot/efi` 挂载分区。请记住，`/boot/efi` 仅支持 RAID 1，需要使用 FAT32 文件系统来格式化该分区。



分区如下所示：

11. 单击接受结束分区程序。

    建议的分区页面上会显示新的建议。

12. 继续安装。对于包含独立 `/boot/efi` 分区的 UEFI 计算机，请在安装设置屏幕上单击引导，然后将引导加载程序设置为 GRUB2 for EFI。检查启用安全引导支持选项是否已激活。

    只要重引导服务器，就会在引导时启动设备映射程序，以便自动识别软件 RAID，并启动根 (/) 分区上的操作系统。

9 创建软件 RAID 10 设备

本章说明如何设置嵌套和复杂 RAID 10 设备。RAID 10 设备包含嵌套 RAID 1（镜像）和 RAID 0（分段）阵列。嵌套 RAID 可以设置为条带化镜像 (RAID 1+0) 或镜像化条带 (RAID 0+1)。复杂 RAID 10 设置支持更高的数据冗余级别，因此镜像与分段兼得，并且拥有更多的数据安全措施。

9.1 使用 mdadm 创建嵌套 RAID 10 设备

嵌套的 RAID 设备由使用另一个 RAID 阵列（而不是物理磁盘）作为其基本元素的 RAID 阵列组成。此配置的目标是提高 RAID 的性能和容错能力。YaST 不支持设置嵌套 RAID 级别，但可以通过 **mdadm** 命令行工具实现。

根据嵌套的顺序，可以设置两个不同的嵌套 RAID。本文使用了下列术语：

- **RAID 1+0：** 先构建 RAID 1（镜像）阵列，然后组合形成 RAID 0（条带化）阵列。
- **RAID 0+1：** 先构建 RAID 0（条带化）阵列，然后组合形成 RAID 1（镜像）阵列。

下表描述嵌套为 1+0 和 0+1 的 RAID 10 的优点和缺点。假定您使用的储存对象驻留在不同的磁盘上，每个都有专用的 I/O 功能。

表 9.1： 嵌套的 RAID 级别

RAID 级别	说明	性能和容错
10 (1+0)	使用 RAID 1（镜像）阵列构建的 RAID 0（条带化）	RAID 1+0 提供高级别的 I/O 性能、数据冗余、和磁盘容错。因为 RAID 0 中的每个成员设备都分别镜像，因此可以容忍多个磁盘故障，并且只要失败的磁盘在不同的镜像中，数据仍然可用。 您可以选择为每个底层的镜像阵列配置一个备用设备，或配置一个备用设备充当服务于所有镜像的备用组。

RAID 级别	说明	性能和容错
10 (0+1)	使用 RAID 0（条带化）阵列构建的 RAID 1（镜像）	RAID 0+1 提供高级别的 I/O 性能和数据冗余，但比 1+0 的容错稍差。如果在镜像的一端多个磁盘失败，则另一个镜像可用。但是，如果在镜像的两端同时丢失磁盘，则所有数据会丢失。 这种解决方案比 1+0 解决方案磁盘容错差，但是如果需要在不同的站点执行维护或维护镜像，则可以使镜像的整个一端脱机，并仍可以有完全正常的储存设备。同时，如果丢失两个站点之间的连接，每个站点会彼此独立地运行。如果条带化镜像分段，则不是这种情况，因为这些镜像在较低级别进行管理。 如果一个设备失败，该端的镜像则会失败，因为 RAID 1 不容错。创建新的 RAID 0 以替换失败的一端，然后重新同步这两个镜像。

9.1.1 使用 mdadm 创建嵌套的 RAID 10 (1+0)

嵌套的 RAID 1+0 的构建方法是，创建两个或更多 RAID 1（镜像），然后使用它们作为 RAID 0 中的组件设备。

重要：多路径

如果需要管理到这些设备的多个连接，则在配置这些 RAID 设备之前，必须配置多路径 I/O。有关信息，请参见第 18 章 “管理设备的多路径 I/O”。

本节中的过程使用下表中显示的设备名。确保使用您自己的设备名称修改这些设备名称。

表 9.2： 通过嵌套创建 RAID 10 (1+0) 的场景

原始设备	RAID 1（镜像）	RAID 1+0（条带化镜像）
`/dev/sdb1`	`/dev/md0`	`/dev/md2`

原始设备	RAID 1（镜像）	RAID 1+0（条带化镜像）
/dev/sdc1		
/dev/sdd1 /dev/sde1	/dev/md1	

1. 打开一个终端控制台。
2. 如果需要，使用 parted 等磁盘分区程序创建四个大小相同的 0xFD Linux RAID 分区。
3. 创建两个软件 RAID 1 设备，为每个 RAID 设备使用两个不同的设备。在命令提示符处，输入以下两个命令：

```
> sudo mdadm --create /dev/md0 --run --level=1 --raid-devices=2 /dev/sdb1 /
dev/sdc1
sudo mdadm --create /dev/md1 --run --level=1 --raid-devices=2 /dev/sdd1 /
dev/sde1
```

4. 创建嵌套的 RAID 1+0 设备。在命令提示符处，使用您在上一步中创建的软件 RAID 1 设备输入以下命令：

```
> sudo mdadm --create /dev/md2 --run --level=0 --chunk=64 \
--raid-devices=2 /dev/md0 /dev/md1
```

   默认大块大小为 64 KB。

5. 在 RAID 1+0 设备 /dev/md2 上创建文件系统，例如 XFS 文件系统：

```
> sudo mkfs.xfs /dev/md2
```

   要使用其他文件系统，请修改该命令。

6. 编辑 /etc/mdadm.conf 文件，如果它不存在，则创建该文件（例如通过运行 **sudo vi /etc/mdadm.conf**）。添加下列行（如果该文件存在，则第一行很可能也已经存在）。

```
DEVICE containers partitions
ARRAY /dev/md0 UUID=UUID
ARRAY /dev/md1 UUID=UUID
```

```
ARRAY /dev/md2 UUID=UUID
```

每个设备的 UUID 可以使用以下命令检索：

```
> sudo mdadm -D /dev/DEVICE | grep UUID
```

7. 编辑 /etc/fstab 文件以添加 RAID 1+0 设备 /dev/md2 的项。下列示例显示了采用 XFS 文件系统并以 /data 做为挂载点的 RAID 设备的项。

```
/dev/md2 /data xfs defaults 1 2
```

8. 挂载 RAID 设备：

```
> sudo mount /data
```

9.1.2 使用 mdadm 创建嵌套的 RAID 10 (0+1)

嵌套的 RAID 0+1 的构建方法是，创建两个到四个 RAID 0（分段）设备，然后将它们镜像为 RAID 1 中的组件设备。

重要：多路径

如果需要管理到这些设备的多个连接，则在配置这些 RAID 设备之前，必须配置多路径 I/O。有关信息，请参见第 18 章 “管理设备的多路径 I/O”。

在此配置中，不能为底层的 RAID 0 设备指定备用设备，因为 RAID 0 不能容忍设备丢失。如果在镜像的一端一个设备失败，则必须创建一个替换 RAID 0 设备，然后将其添加到镜像。

本节中的过程使用下表中显示的设备名。确保使用您自己的设备名称修改这些设备名称。

表 9.3： 通过嵌套创建 RAID 10 (0+1) 的场景

原始设备	RAID 0（条带化）	RAID 0+1（镜像条带化）
/dev/sdb1 /dev/sdc1	/dev/md0	/dev/md2
/dev/sdd1	/dev/md1	

原始设备	RAID 0（条带化）	RAID 0+1（镜像条带化）
/dev/sde1		

1. 打开一个终端控制台。
2. 如果需要，使用 parted 等磁盘分区程序创建四个大小相同的 0xFD Linux RAID 分区。
3. 创建两个软件 RAID 0 设备，为每个 RAID 0 设备使用两个不同的设备。在命令提示符处，输入以下两个命令：

   ```
   > sudo mdadm --create /dev/md0 --run --level=0 --chunk=64 \
   --raid-devices=2 /dev/sdb1 /dev/sdc1
   sudo mdadm --create /dev/md1 --run --level=0 --chunk=64 \
   --raid-devices=2 /dev/sdd1 /dev/sde1
   ```

   默认大块大小为 64 KB。
4. 创建嵌套的 RAID 0+1 设备。在命令提示符处，使用您在上一步中创建的软件 RAID 0 设备输入以下命令：

   ```
   > sudo mdadm --create /dev/md2 --run --level=1 --raid-devices=2 /dev/md0 /
   dev/md1
   ```
5. 在 RAID 1+0 设备 /dev/md2 上创建文件系统，例如 XFS 文件系统：

   ```
   > sudo mkfs.xfs /dev/md2
   ```

   要使用其他文件系统，请修改该命令。
6. 编辑 /etc/mdadm.conf 文件，如果它不存在，则创建该文件（例如通过运行 **sudo vi /etc/mdadm.conf**）。添加下列行（如果该文件存在，第一行很可能也已经存在）。

   ```
   DEVICE containers partitions
   ARRAY /dev/md0 UUID=UUID
   ARRAY /dev/md1 UUID=UUID
   ARRAY /dev/md2 UUID=UUID
   ```

   每个设备的 UUID 可以使用以下命令检索：

```
> sudo mdadm -D /dev/DEVICE | grep UUID
```

7. 编辑 `/etc/fstab` 文件以添加 RAID 1+0 设备 `/dev/md2` 的项。下列示例显示了采用 XFS 文件系统并以 `/data` 做为挂载点的 RAID 设备的项。

```
/dev/md2 /data xfs defaults 1 2
```

8. 挂载 RAID 设备：

```
> sudo mount /data
```

9.2 创建复杂 RAID 10

YaST（以及带 `--level=10` 选项的 **mdadm**）可创建单一复杂软件 RAID 10，它结合了 RAID 0（分段）与 RAID 1（镜像）的功能。所有数据块的多个副本遵循一个分段准则在多个驱动器上排列。组件设备应大小相同。

复杂 RAID 10 与嵌套 RAID 10 (1+0) 的目的类似，但在以下方面不同：

表 9.4： 复杂 RAID 10 与嵌套 RAID 10 的比较

功能	复杂 RAID 10	嵌套 RAID 10 (1+0)
设备数	允许组件设备数为奇数或偶数	要求组件设备数为偶数
组件设备	作为单个 RAID 设备管理	作为嵌套 RAID 设备管理
分段	在组件设备的近布局或远布局中发生分段。 远布局提供根据驱动器数（而不是 RAID 1 对数）缩放的串行读吞吐量。	在组件设备之间连续发生分段
数据的多个副本	两个或更多副本，最多为阵列中的设备数	每个镜像段上的副本

功能	复杂 RAID 10	嵌套 RAID 10 (1+0)
热备用设备	单个备用设备可以服务于所有组件设备	为每个底层的镜像阵列配置一个备用设备，或配置一个备用设备充当服务于所有镜像的备用组。

9.2.1 复杂 RAID 10 中的设备和复本数

配置复杂 RAID 10 阵列时，必须指定需要的每个数据块的复本数。默认复本数是 2，但该值可以是 2 到阵列中设备数之间的任何数字。

必须至少使用指定的复本数的组件设备。但是，RAID 10 阵列中的组件设备数不必是每个数据块的复本数的倍数。有效储存大小是设备数除以复本数。

例如，如果为使用 5 个组件设备创建的阵列指定 2 个复本，则每个块的副本储存在两个不同的设备上。所有数据的一个副本的有效储存大小是 5/2 或 2.5 乘以组件设备的大小。

9.2.2 布局

复杂 RAID 10 设置支持 3 种不同的布局，这些布局定义了在磁盘上排列数据块的方式。可用布局有近（默认值）、远和偏移。它们的性能特性各不相同，因此您必须选择适合自己工作负载的布局。

9.2.2.1 近布局

使用近布局，数据块的副本会在不同的组件设备上彼此邻近地条带化。即，一个数据块的多个副本在不同设备中的偏移类似。近布局是 RAID 10 的默认布局。例如，如果使用奇数个组件设备以及数据的两个副本，则一些副本可能在设备的一个大块中。

在半数的驱动器上，复杂 RAID 10 的近布局在读写性能上与 RAID 0 类似。

具有偶数个磁盘和两个复本的近布局：

```
sda1 sdb1 sdc1 sde1
```

```
  0    0    1    1
  2    2    3    3
  4    4    5    5
  6    6    7    7
  8    8    9    9
```

具有奇数个磁盘和两个复本的近布局：

```
sda1 sdb1 sdc1 sde1 sdf1
  0    0    1    1    2
  2    3    3    4    4
  5    5    6    6    7
  7    8    8    9    9
  10   10   11   11   12
```

9.2.2.2 远布局

远布局在所有驱动器的前面部分条带化数据，然后在所有驱动器的后面部分条带化数据的另一份副本，以确保块的所有副本在不同的驱动器上。第二个值集合在组件驱动器的中间开始。

使用远布局，复杂 RAID 10 的读取性能类似于所有驱动器上的 RAID 0，但写入速度比 RAID 0 慢得多，因为前者需要更多地搜寻驱动器头。适用于读密集性操作，例如只读文件服务器。

RAID 10 的写入速度类似于其他镜像 RAID 类型（例如使用近布局的 RAID 1 和 RAID 10），因为该文件系统的电梯式操作会以最佳方式而不是原始写入方式调度写入。使用远布局的 RAID 10 最适合镜像写入应用程序。

具有偶数个磁盘和两个复本的远布局：

```
sda1 sdb1 sdc1 sde1
  0    1    2    3
  4    5    6    7
  . . .
  3    0    1    2
  7    4    5    6
```

具有奇数个磁盘和两个复本的远布局：

```
sda1 sdb1 sdc1 sde1 sdf1
```

```
  0    1    2    3    4
  5    6    7    8    9
  . . .
  4    0    1    2    3
  9    5    6    7    8
```

9.2.2.3 偏移布局

偏移布局会复制分段，从而使指定大块的多个副本在连续的驱动器上以连续偏移进行布局。事实上，会复制每个分段且副本会按每个设备进行偏移。如果使用适当较大的大块大小且不超过写入的寻道大小，则此方式会赋予远布局类似的读特征。

具有偶数个磁盘和两个复本的偏移布局：

```
sda1 sdb1 sdc1 sde1
  0    1    2    3
  3    0    1    2
  4    5    6    7
  7    4    5    6
  8    9   10   11
 11    8    9   10
```

具有奇数个磁盘和两个复本的偏移布局：

```
sda1 sdb1 sdc1 sde1 sdf1
  0    1    2    3    4
  4    0    1    2    3
  5    6    7    8    9
  9    5    6    7    8
 10   11   12   13   14
 14   10   11   12   13
```

9.2.2.4 使用 YaST 和 mdadm 指定复本数和布局

复本数和布局在 YaST 中或使用 mdadm 的 `--layout` 参数指定为奇偶校验算法。接受的值如下：

nN

对近布局指定 n，并用复本数替换 N。n2 是默认值，在未设置布局和复本数时使用。

fN

对远布局指定 f，并用复本数替换 N。

oN

对偏移布局指定 o，并用复本数替换 N。

注意：复本数

YaST 会自动提供奇偶校验算法参数的所有可能的值供您选择。

9.2.3 使用 YaST 分区程序创建复杂 RAID 10

1. 启动 YaST 并打开分区程序。
2. 如果需要，请创建应该与 RAID 配置搭配使用的分区。请勿将它们格式化，并将分区类型设置为 0xFD Linux RAID。使用现有分区时，不需要更改它们的分区类型 — YaST 会自动更改。有关细节，请参考《部署指南》, 第 10 章 “专家分区程序”, 第 10.1 节 “使用专家分区程序”。

   对于 RAID 10，至少需要四个分区。强烈建议您使用储存在不同硬盘上的分区，以降低当其中一个损坏时遗失数据的风险。建议仅使用大小相同的分区，因为每个段仅可将相同的空间量作为最小大小的分区。
3. 在左侧面板中，选择 RAID。

   右侧面板中即会打开现有 RAID 配置的列表。
4. 在 RAID 页面的左下方，单击 添加 RAID。
5. 在 RAID 类型下，选择 RAID 10（镜像和分段）。

   您可以选择性地为 RAID 指派一个 RAID 名称。这样，RAID 的名称将是 /dev/md/NAME。有关更多信息，请参见第 7.2.1 节 “RAID 名称”。
6. 在可用设备列表中选择所需的 分区，然后单击添加，将其移到所选设备列表中。

7. （可选）单击分类，指定各磁盘在 RAID 阵列中的首选顺序。

   对于容易受到磁盘添加顺序影响的 RAID 类型（例如 RAID 10），您可以指定以何种顺序使用设备。这将确保一半的阵列在一个磁盘子系统上，另一半阵列在另一个磁盘子系统上。例如，如果一个磁盘子系统出现故障，则系统仍可以通过第二个磁盘子系统继续运行。

   a. 依次选择每个磁盘，并单击其中一个 X 类按钮，其中 X 表示要指派给磁盘的字母。可用的类有 A、B、C、D 和 E，但很多情况下需要的类会比较少（例如，仅需要 A 和 B）。以此方式指派所有可用的 RAID 磁盘。

      可以按 `Ctrl` 或 `Shift` 键选择多个设备。也可以右键单击所选设备并从环境菜单中选择合适的类。

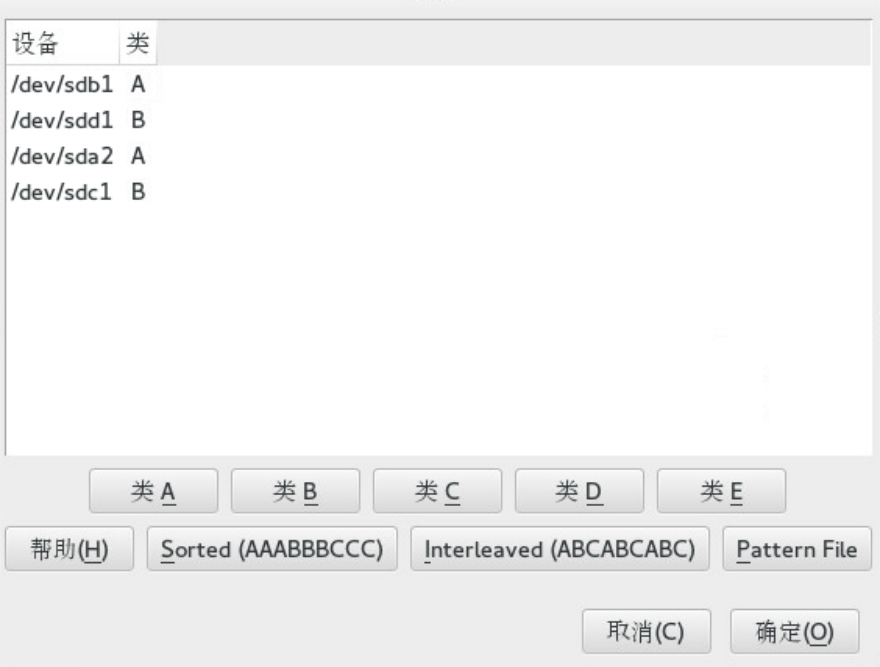
   b. 选择以下其中一个排序选项指定设备的顺序：

      排序： 将所有 A 类设备排序在所有 B 类设备之前，并以此类推。例如：AABBCC。

      交错： 按照先是 A 类的第一个设备，然后是 B 类的第一个设备，以此类推直到排完所有指派了后面类的设备。再然后是 A 类的第二个设备，B 类的第二个设备，以此类推。没有类的所有设备将排在设备列表的末尾。例如：ABCABC。

模式文件： 选择一个包含多行的现有文件，其中的每一行都是一个正则表达式以及一个类名称 (`"sda.* A"`).所有与该正则表达式匹配的设备都将被指派给该行指定的类。正则表达式会依次与内核名称 (`/dev/sda1`)、udev 路径名称 (`/dev/disk/by-path/pci-0000:00:1f.2-scsi-0:0:0:0-part1`) 和 udev ID (`dev/disk/by-id/ata-ST3500418AS_9VMN8X8L-part1`) 进行比对。如果设备名称与多个正则表达式匹配，则会使用首个匹配项来决定类。

c. 在对话框的底部，单击确定，以确认顺序。



8. 单击下一步。

9. 在 RAID 选项下，指定大块大小和奇偶校验算法，然后单击下一步。
   对于 RAID 10，奇偶校验选项包括 n（近）、f（远）和 o（偏移）。数字表示所需的每个数据块的复本数。默认值为 2。有关信息，请参见第 9.2.2 节 “布局”。

10. 将文件系统和挂载选项添加至 RAID 设备，然后单击完成。

11. 单击下一步。

12. 确认要进行的更改，然后单击完成以创建 RAID。

9.2.4 使用 mdadm 创建复杂 RAID 10

本节中的过程使用下表中显示的设备名。确保使用您自己的设备名称修改这些设备名称。

表 9.5： 使用 MDADM 创建 RAID 10 的场景

原始设备	RAID 10
/dev/sdf1 /dev/sdg1 /dev/sdh1 /dev/sdi1	/dev/md3

1. 打开一个终端控制台。
2. 如果需要，使用 parted 等磁盘分区程序创建至少四个大小相同的 0xFD Linux RAID 分区。
3. 输入以下命令创建 RAID 10。

   ```
   > sudo mdadm --create /dev/md3 --run --level=10 --chunk=32 --raid-devices=4
    \
   /dev/sdf1 /dev/sdg1 /dev/sdh1 /dev/sdi1
   ```

   请务必根据您的设置调整 --raid-devices 的值和分区列表。
   该命令会创建采用近布局并有两个复本的阵列。要更改这两个值中的任何一个，请按第 9.2.2.4 节 “使用 YaST 和 mdadm 指定复本数和布局” 中所述使用 --layout。
4. 在 RAID 10 设备 /dev/md3 上创建文件系统，例如 XFS 文件系统：

   ```
   > sudo mkfs.xfs /dev/md3
   ```

   要使用其他文件系统，请修改该命令。
5. 编辑 /etc/mdadm.conf 文件，如果它不存在，则创建该文件（例如通过运行 **sudo vi /etc/mdadm.conf**）。添加下列行（如果该文件存在，第一行很可能也已经存在）。

```
DEVICE containers partitions
ARRAY /dev/md3 UUID=UUID
```

设备的 UUID 可以使用以下命令检索：

```
> sudo mdadm -D /dev/md3 | grep UUID
```

6. 编辑 `/etc/fstab` 文件以添加 RAID 10 设备 `/dev/md3` 的项目。下列示例显示了采用 XFS 文件系统并以 `/data` 做为挂载点的 RAID 设备的项。

```
/dev/md3 /data xfs defaults 1 2
```

7. 挂载 RAID 设备：

```
> sudo mount /data
```

10 创建降级 RAID 阵列

降级阵列是缺少某些设备的阵列。仅对 RAID 1、RAID 4、RAID 5 和 RAID 6 支持降级阵列。这些 RAID 类型被设计为作为容错功能的一部分，容忍缺少一些设备。通常，当一个设备失败时，会发生降级阵列。可以故意创建一个降级阵列。

RAID 类型	允许缺少的槽数
RAID 1	除一个设备外的所有设备
RAID 4	一个槽
RAID 5	一个槽
RAID 6	一个或两个槽

要创建缺少某些设备的降级阵列，只需使用单词 `missing` 替换设备名即可。这会导致 **`mdadm`** 将阵列中的相应槽保留为空。

当创建 RAID 5 阵列时，**`mdadm`** 会自动使用额外的备用驱动器创建降级阵列。这是因为将备用设备构建为降级阵列通常比在非降级但不清洁的阵列上重新同步奇偶校验要更快。您可以使用 `--force` 选项覆盖该功能。

在要创建 RAID，但要使用的一个设备上已经有数据时，创建降级阵列非常有用。在这种情况下，使用其他设备创建一个降级阵列，将使用中的设备上的数据复制到以降级模式运行的 RAID 上，将该设备添加到 RAID 中，然后等待 RAID 重构建以便该数据分散在所有设备中。以下过程是如此处理的一个示例：

1. 若要使用单个驱动器 `/dev/sd 1` 创建降级的 RAID 1 设备 `/dev/md0`，请在命令提示符处输入以下命令：

   ```
   > sudo mdadm --create /dev/md0 -l 1 -n 2 /dev/sda1 missing
   ```

   该设备的大小应等于或大于计划添加到该设备的设备。

2. 如果要添加到镜像的设备包含要移到 RAID 阵列中的数据，现在将其复制到处于降级模式的 RAID 阵列中。

3. 将您从中复制数据的设备添加至镜像。例如，要向 RAID 中添加 `/dev/sdb1`，在命令提示符处输入以下命令：

```
> sudo mdadm /dev/md0 -a /dev/sdb1
```

您一次只能添加一个设备。必须等到内核构建镜像并将其完全联机，才能添加另一个镜像。

4. 在命令提示符处输入以下命令，监视构建过程：

```
> sudo cat /proc/mdstat
```

要在每秒刷新重建进度时查看此进度，请输入

```
> sudo watch -n 1 cat /proc/mdstat
```

11 使用 mdadm 调整软件 RAID 阵列的大小

本章描述如何使用多设备管理 (**mdadm(8)**) 工具增加或减小软件 RAID 1、4、5 或 6 设备的大小。

调整现有软件 RAID 设备的大小涉及增加或降低每个组件分区提供的空间。在 RAID 上驻留的文件系统也必须能够调整大小，以充分利用设备上可用空间的更改。在 SUSE Linux Enterprise Server 中，文件系统重设置大小实用程序可用于 Btrfs、Ext2、Ext3、Ext4、 和 XFS 文件系统（仅限增加大小）。有关更多信息，请参考第 2 章 “调整文件系统的大小”。

mdadm 工具仅支持调整软件 RAID 级别 1、4、5 和 6 的大小。这些 RAID 级别提供磁盘容错，这样在调整大小时，可以一次卸下一个组件分区。基本上来说，可以对 RAID 分区执行热调整大小，但是这样做时，必须额外注意您的数据。

警告：调整大小之前请备份数据

调整任何分区或文件系统的大小涉及可能会导致数据丢失的风险。为了避免数据丢失，请确保在开始任何调整大小任务之前备份您的数据。

调整 RAID 大小涉及以下任务。执行这些任务的顺序取决于是增加还是减少大小。

表 11.1： 调整 RAID 大小中涉及的任务

任务	说明	增加大小的顺序	减小大小的顺序
调整每个组件分区的大小。	增加或减小每个组件分区的活动大小。一次仅可删除一个组件分区，修改其大小，然后将其返回到 RAID。	1	2
调整软件 RAID 本身的大小。	RAID 不会自动知道您对底层组件分区大小进行的增加或减小操作。您必须向其告知新的大小。	2	3
调整文件系统的大小。	必须调整驻留在 RAID 上的文件系统的大小。此操作只适用于提供了用于重设置大小工具的文件系统。	3	1

下列各部分中的程序使用在下表中所示的设备名称。确保使用您自己的设备名称修改这些名称。

表 11.2： 增加组件分区的大小的场景

RAID 设备	组件分区
/dev/md0	/dev/sda1 /dev/sdb1 /dev/sdc1

11.1 增加软件 RAID 的大小

增大软件 RAID 的大小涉及按给定顺序执行下列任务：增加所有组成 RAID 的所有分区的大小，增加 RAID 本身的大小，最后增加文件系统的大小。

警告：潜在数据丢失

如果 RAID 没有磁盘容错，或只是不一致，则在删除其任何分区的情况下，将会导致数据丢失。删除分区时要非常小心，并确保有可用的数据备份。

11.1.1 增加组件分区的大小

应用本节中的过程以增加 RAID 1、4、5 或 6 的大小。对于 RAID 中的每个组件分区，从 RAID 中删除该分区，修改其大小，将其返回到 RAID，然后等待 RAID 稳定下来以继续。删除一个分区时，RAID 会以降级模式运行，没有磁盘容错或降低磁盘容错。即使对于可以容忍多个并行磁盘故障的 RAID，也不要一次删除多个组件分区。若要增加 RAID 组件分区的大小，请执行下列步骤：

1. 打开一个终端控制台。
2. 通过输入以下命令，确保 RAID 阵列一致并同步

```
> cat /proc/mdstat
```

如果根据此命令的输出，RAID 阵列仍然正在同步，则必须等到同步完成，才能继续。

3. 从 RAID 阵列中删除一个组件分区。例如，要删除 /dev/sda1，请输入

```
> sudo mdadm /dev/md0 --fail /dev/sda1 --remove /dev/sda1
```

为确保操作成功，必须指定失败和去除操作。

4. 执行下列操作之一，增加在上一步中去除的分区的大小：

   - 使用磁盘分区程序（例如 YaST 分区程序）或命令行工具 parted 增加分区的大小。该选项是通常的选项。
   - 将分区驻留的磁盘替换为高容量设备。仅当系统不访问原始磁盘上的任何其他文件系统时，该选项才可用。当将替换设备添加回 RAID 时，同步数据需要的时间长得多，因为原始设备上的所有数据都必须重构建。

5. 将该分区重新添加到 RAID 阵列。例如，要添加 /dev/sda1，请输入

```
> sudo mdadm -a /dev/md0 /dev/sda1
```

等到 RAID 同步并一致，然后再继续下一个分区。

6. 对阵列中的每个剩余组件设备重复执行这些步骤。确保按照正确的组件分区修改命令。
7. 如果得到一个消息，告知您内核不能重读 RAID 的分区表，则必须在调整所有分区大小后重引导计算机，以强制更新分区表。
8. 继续执行第 11.1.2 节 “增加 RAID 阵列的大小”。

11.1.2 增加 RAID 阵列的大小

调整 RAID 中的每个组件分区之后（请参见第 11.1.1 节 “增加组件分区的大小”），RAID 阵列配置将继续使用原始阵列大小，直到您强制其了解新的可用空间。您可以为 RAID 指定大小或使用最大可用空间。

本节中的过程对 RAID 设备使用设备名 /dev/md0。确保使用您自己的设备名称修改名称。

1. 打开一个终端控制台。

2. 通过输入以下命令，确保 RAID 阵列一致并同步

```
> cat /proc/mdstat
```

如果根据此命令的输出，RAID 阵列仍然正在同步，则必须等到同步完成，才能继续。

3. 通过输入以下命令，检查阵列了解到的阵列的÷大小和设备大小

```
> sudo mdadm -D /dev/md0 | grep -e "Array Size" -e "Dev Size"
```

4. 执行以下操作之一：

   - 通过输入以下命令，将阵列大小增加到最大可用大小

     ```
     > sudo mdadm --grow /dev/md0 -z max
     ```

   - 通过输入以下命令，将阵列大小增加到最大可用大小

     ```
     > sudo mdadm --grow /dev/md0 -z max --assume-clean
     ```

     阵列会使用已添加到设备中的任何空间，但不会同步此空间。建议对 RAID 1 使用此命令，因为该级别不需要同步。如果添加到成员设备中的空间已预先置零，则对其他 RAID 级别可能也有用。

   - 通过输入以下命令，将阵列大小增加到指定值

     ```
     > sudo mdadm --grow /dev/md0 -z SIZE
     ```

     将 `SIZE` 替换为表示所需大小（KB，每 KB 等于 1024 字节）的整数值。

5. 通过输入以下命令，重新检查阵列了解到的阵列大小和设备大小

```
> sudo mdadm -D /dev/md0 | grep -e "Array Size" -e "Dev Size"
```

6. 执行以下操作之一：

   - 如果已成功调整阵列大小，请继续第 11.1.3 节 “增加文件系统的大小”。

   - 如果未能和预期一样调整阵列大小，则请重引导，然后重试该过程。

11.1.3 增加文件系统的大小

增加阵列大小之后（请参见第 11.1.2 节 “增加 RAID 阵列的大小”），您就准备好调整文件系统大小了。

您可以将文件系统的大小增加到最大可用空间或指定精确大小。为文件系统指定精确大小时，请确保新大小满足以下条件：

- 新大小必须大于现有数据的大小；否则会发生数据丢失。
- 新大小必须等于或小于当前 RAID 大小，因为文件系统大小不能超出可用空间。

有关详细，请参见第 2 章 “调整文件系统的大小”。

11.2 减小软件 RAID 的大小

减少软件 RAID 的大小涉及按顺序完成下列任务：减少文件系统的大小，减少所有组成分区 RAID 的大小，最后减少 RAID 本身的的大小。

警告：潜在数据丢失

如果 RAID 没有磁盘容错，或只是不一致，则在删除其任何分区的情况下，将会导致数据丢失。删除分区时要非常小心，并确保有可用的数据备份。

重要：XFS

XFS 格式文件系统的大小无法减少，因为 XFS 不支持此功能。因此，不能减少使用 XFS 文件系统的 RAID 的大小。

11.2.1 减小文件系统的大小

当减小 RAID 设备上的文件系统的大小时，请确保新的大小满足以下条件：

- 新大小必须大于现有数据的大小；否则会发生数据丢失。
- 新大小必须等于或小于当前 RAID 大小，因为文件系统大小不能超出可用空间。

有关详细，请参见第 2 章 “调整文件系统的大小”。

11.2.2 减小 RAID 阵列的大小

调整文件系统的大小（请参见第 11.2.1 节 “减小文件系统的大小”）之后，RAID 阵列配置会继续使用其原始阵列大小，直到您强制它减少可用空间。使用 **mdadm --grow** 模式强制 RAID 使用较小的段大小。为此，您必须使用 -z 选项指定 RAID 中的每个设备上可使用的空间量（单位为 KB）。此大小必须是大块大小的倍数，且必须为将要写入设备的 RAID 超块预留大约 128KB 的空间。

本节中的过程对 RAID 设备使用设备名 /dev/md0。确保使用您自己的设备名称修改这些命令。

1. 打开一个终端控制台。
2. 通过输入以下命令，检查阵列了解到的阵列大小和设备大小

   ```
   > sudo mdadm -D /dev/md0 | grep -e "Array Size" -e "Dev Size"
   ```

3. 输入以下命令将阵列的设备大小减少至指定值

   ```
   > sudo mdadm --grow /dev/md0 -z SIZE
   ```

   将 SIZE 替换为表示所需大小的整数值（单位为 KB）。（1 KB 是 1024 字节。）
   例如，以下命令将每个 RAID 设备的段大小设置为大约 40 GB，其中大块大小为 64 KB。还包含为 RAID 超块预留的 128 KB。

   ```
   > sudo mdadm --grow /dev/md2 -z 41943168
   ```

4. 通过输入以下命令，重新检查阵列了解到的阵列大小和设备大小

   ```
   > sudo mdadm -D /dev/md0 | grep -e "Array Size" -e "Device Size"
   ```

5. 执行以下操作之一：
   - 如果已成功调整阵列大小，请继续第 11.2.3 节 “减小组件分区的大小”。
   - 如果未能和预期一样调整阵列大小，则请重引导，然后重试该过程。

11.2.3 减小组件分区的大小

减小 RAID 中每个设备使用的段的大小之后（请参见第 11.2.2 节 “减小 RAID 阵列的大小”），RAID 将不会使用每个组件分区的剩余空间。您可以让分区保持其当前大小，以为 RAID 将来的增长留出空间，或者也可以回收这些当前未使用的空间。

要回收空间，请逐个减少组件分区。对于每个组件分区执行以下步骤：从 RAID 中删除它，减小其分区大小，将该分区装回到 RAID，然后等到 RAID 稳定。要允许元数据，则应指定比您在第 11.2.2 节 “减小 RAID 阵列的大小”中为 RAID 指定的大小略大的大小值。

删除一个分区时，RAID 会以降级模式运行，没有磁盘容错或降低磁盘容错。即使对于可以容忍多个并行磁盘故障的 RAID，也不要一次删除多个组件分区。若要减小 RAID 组件分区的大小，请执行下列步骤：

1. 打开一个终端控制台。
2. 通过输入以下命令，确保 RAID 阵列一致并同步

   ```
   > cat /proc/mdstat
   ```

   如果根据此命令的输出，RAID 阵列仍然正在同步，则必须等到同步完成，才能继续。
3. 从 RAID 阵列中删除一个组件分区。例如，要删除 `/dev/sda1`，请输入

   ```
   > sudo mdadm /dev/md0 --fail /dev/sda1 --remove /dev/sda1
   ```

   为确保操作成功，必须指定失败和去除操作。
4. 减小在上一步中去除的分区的大小，让其值略大于为段设置的大小。该大小应是大块大小的倍数，并为 RAID 超块预留 128 KB 的空间。使用磁盘分区程序（例如 YaST 分区程序）或命令行工具 parted 减少分区的大小。
5. 将该分区重新添加到 RAID 阵列。例如，要添加 `/dev/sda1`，请输入

   ```
   > sudo mdadm -a /dev/md0 /dev/sda1
   ```

   等到 RAID 同步并一致，然后再继续下一个分区。
6. 对阵列中的每个剩余组件设备重复执行这些步骤。确保按照正确的组件分区修改命令。

7. 如果得到一个消息，告知您内核无法重读 RAID 的分区表，则您必须在重新调整其所有组件分区大小后重引导计算机。

8. （可选）扩展 RAID 和文件系统的大小，以使用当前较小组件分区中的最大空间量，并在此后增加文件系统的大小。有关指导，请参见第 11.1.2 节 “增加 RAID 阵列的大小”。

12 适用于 MD 软件 RAID 的储存机箱 LED 实用程序

储存机箱 LED 监视实用程序 (**ledmon**) 和 LED 控件 (**ledctl**) 实用程序属于 Linux 用户空间应用程序，它们使用多种接口和协议来控制储存机箱 LED。主要用途是视觉化显示使用 mdadm 实用程序创建的 Linux MD 软件 RAID 设备的状态。ledmon 守护程序会监视驱动器阵列的状态，然后相应更新驱动器 LED 的状态。您可以使用 **ledctl** 实用程序为指定的设备设置 LED 模式。

这些 LED 实用程序使用 SGPIO（通用串行输入/输出）规范（小型 (SFF) 8485）以及 SCSI 机箱服务 (SES) 2 协议控制 LED。它们实施 SGPIO 的 SFF-8489 规范中的国际闪烁模式解释 (IBPI) 模式。IBPI 定义了 SGPIO 标准将如何解释成驱动器以及底板上各插槽的状态，以及底板将如何通过 LED 视觉化显示状态。

有些储存机箱并未严格遵循 SFF-8489 规范。机箱处理器可能会接受 IBPI 模式，但不会按照 SFF-8489 规范闪烁 LED，或者处理器可能仅支持部分 IBPI 模式。

ledmon 以及 **ledctl** 实用程序不支持 LED 管理(AHCI) 和 SAF-TE 协议。

ledmon 和 **ledctl** 应用程序经验证可以用于 Intel AHCI 控制器以及 Intel SAS 控制器等 Intel 储存控制器。它们还支持使用 PCIe-SSD（固态硬盘）机箱 LED 来控制属于 MD 软件 RAID 卷一部分的 PCIe-SSD 设备的储存机箱状态（正常、故障、正在重建）LED。这些应用程序也可能可以用于其他供应商提供的符合 IBPI 的储存控制器（特别是 SAS/SCSI 控制器）；不过其他供应商的控制器未经过测试。

ledmon 和 **ledctl** 都属于 ledmon 软件包，该包在默认状态下不予安装。执行 **sudo zypper in ledmon** 可安装它。

12.1 储存机箱 LED 监视服务

ledmon 应用程序是一个守护程序进程，它会持续监视 MD 软件 RAID 设备的状态或储存机箱或驱动器机架中块设备的状态。一次只能运行一个该守护程序的实例。ledmon 守护程序是 Intel 机箱 LED 实用程序的一部分。

状态通过与储存阵列机箱或驱动器机架中的每个槽对应的 LED 视觉化显示。应用程序会监视所有软件 RAID 设备并视觉化显示其状态。无法通过该应用程序仅监视选定的软件 RAID 卷。

ledmon 守护程序支持两类 LED 系统：双 LED 系统（活动 LED 和状态 LED）与三 LED 系统（活动 LED、查找 LED 和故障 LED）。此工具在访问 LED 时具有最高优先级。

要启动 ledmon，请输入

```
> sudo ledmon [options]
```

其中 [options] 是下列一或多项：

ledmon 的选项

-c 路径，
--confg=路径

配置从 ~/.ledctl 或 /etc/ledcfg.conf（如果存在）中读取。使用此选项来指定替代的配置文件。

目前此选项无效，因为尚未实施对配置文件的支持。有关细节，请参见 **man 5 ledctl.conf**。

-l 路径，
--log=路径

设置本地日志文件的路径。如果指定此用户定义的文件，则将不会使用全局日志文件 /var/log/ledmon.log。

-t 秒数，
--interval=秒数

设置 sysfs 的扫描间隔时间。该值以秒为单位。最小值为 5 秒。不指定最大值。

--quiet、--error、--warning、--info、--debug、--all

指定冗长级别。级别选项按从无信息到最详细信息的顺序指定。如果不记录日志，则使用 --quiet 选项。如果要记录所有内容，则使用 --all 选项。如果指定多个冗长选项，则会应用命令中的最后一个选项。

-h，
--help

将命令信息打印至控制台，然后退出。

`-v`，
`--version`

显示 **ledmon** 的版本以及许可证的相关信息，然后退出。

注意：已知问题

ledmon 守护程序无法识别 SFF-8489 规范中的 PFA（故障预警分析）状态。因此无法视觉化显示 PFA 模式。

12.2 储存机箱 LED 控制应用程序

机箱 LED 应用程序 (**ledctl**) 是一款用户空间应用程序，用于控制与储存机箱或驱动器机架中每个槽相关联的 LED。**ledctl** 应用程序是 Intel 机箱 LED 实用程序的组成部分。

在您发出命令时，指定设备的 LED 将被设置为指定的模式，并且所有其他的 LED 都将关闭。运行此应用程序需要 `root` 权限。由于 `ledmon` 应用程序在访问 LED 时具有最高优先级，因此如果 `ledmon` 守护程序正在运行，则由 ledctl 设置的某些模式可能不起作用（“查找”模式除外）。

ledctl 应用程序支持两类 LED 系统：二 LED 系统（活动 LED 和状态 LED）以及三 LED 系统（活动 LED、故障 LED 和查找 LED）。

若要启动 **ledctl**，请输入

```
> sudo [options] PATTERN_NAME=list_of_devices
```

其中 [options] 是下列一或多项：

`-c 路径`，
`--confg=路径`

设置本地配置文件的路径。如果指定此选项，则全局配置文件和用户配置文件都将失效。

-l `路径`，
--log=`路径`

设置本地日志文件的路径。如果指定此用户定义的文件，则将不会使用全局日志文件 `/var/log/ledmon.log`。

`--quiet`

关闭所有向 `stdout` 或从 `stderr` 发送的消息。但在本地文件和 `syslog` 功能中仍会记录这些消息。

`-h`,
`--help`

将命令信息打印至控制台，然后退出。

`-v`,
`--version`

显示 **`ledctl`** 的版本以及许可证的相关信息，然后退出。

12.2.1 模式名称

根据 SFF-8489 规范，**`ledctl`** 应用程序会接受以下 pattern_name 参数的名称。

`locate`

打开与指定设备或空槽相关的查找 LED。此状态用于标识槽或驱动器。

`locate_off`

关闭与指定设备或空槽相关的查找 LED。

`normal`

关闭与指定设备相关的状态 LED、故障 LED 以及查找 LED。

`off`

仅关闭与指定设备相关的状态 LED 和故障 LED。

`ica`,
`degraded`

可视化显示 `In a Critical Array` 模式。

`rebuild`,
`rebuild_p`

可视化显示 `Rebuild` 模式。由于兼容性和旧版原因，支持两种重构建状态。

`ifa`,
`failed_array`
: 可视化显示 `In a Failed Array` 模式。

`hotspare`
: 可视化显示 `Hotspare` 模式。

`pfa`
: 可视化显示 `Predicted Failure Analysis` 模式。

`failure`,
`disk_failed`
: 可视化显示 `Failure` 模式。

ses_abort
: SES-2 R/R ABORT

`ses_rebuild`
: SES-2 REBUILD/REMAP

`ses_ifa`
: SES-2 IN FAILED ARRAY

`ses_ica`
: SES-2 IN CRITICAL ARRAY

`ses_cons_check`
: SES-2 CONS CHECK

`ses_hotspare`
: SES-2 HOTSPARE

`ses_rsvd_dev`
: SES-2 RSVD DEVICE

`ses_ok`
: SES-2 OK

`ses_ident`
: SES-2 IDENT

`ses_rm`
: SES-2 REMOVE

`ses_insert`
: SES-2 INSERT

`ses_missing`
: SES-2 MISSING

`ses_dnr`
: SES-2 DO NOT REMOVE

`ses_active`
: SES-2 ACTIVE

`ses_enable_bb`
: SES-2 ENABLE BYP B

`ses_enable_ba`
: SES-2 ENABLE BYP A

`ses_devoff`
: SES-2 DEVICE OFF

`ses_fault`
: SES-2 FAULT

将非 SES-2 模式发送至机箱中的设备时，该模式会自动转换为上面所示的 SCSI 机箱服务 (SES) 2 模式。

表 12.1： 非 SES-2 模式与 SES-2 模式之间的转换

非 SES-2 模式	SES-2 模式
locate	ses_ident

非 SES-2 模式	SES-2 模式
locate_off	ses_ident
normal	ses_ok
off	ses_ok
ica	ses_ica
degraded	ses_ica
rebuild	ses_rebuild
rebuild_p	ses_rebuild
ifa	ses_ifa
failed_array	ses_ifa
hotspare	ses_hotspare
pfa	ses_rsvd_dev
failure	ses_fault
disk_failed	ses_fault

12.2.2 设备列表

在您发出 **`ledctl`** 命令时，指定设备的 LED 将被设置为指定的模式，并且所有其他的 LED 都将关闭。设备列表可采用以下两种格式中的一种提供：

- 以逗号分隔且无空格的设备列表
- 以空格分隔且用花括号括住的设备列表

如果在同一个命令中指定多个模式，则每一个模式的设备列表可以使用相同格式也可以使用不同格式。有关显示两种列表格式的示例，请参见第 12.2.3 节 “示例”。

设备就是指向 /dev 目录或 /sys/block 目录中文件的路径。该路径可以标识块设备、MD 软件 RAID 设备或容器设备。对于软件 RAID 设备或容器设备，会为所有关联的块设备设置所报告的 LED 状态。

list_of_devices 中列出设备的 LED 将被设置为指定模式 pattern_name 并且所有其他 LED 都将关闭。

12.2.3 示例

查找单个块设备：

```
> sudo ledctl locate=/dev/sda
```

若要关闭单个块设备的“查找 LED”，请执行以下步骤：

```
> sudo ledctl locate_off=/dev/sda
```

查找 MD 软件 RAID 设备的磁盘，并同时为两个块设备设置重构建模式：

```
> sudo ledctl locate=/dev/md127 rebuild={ /sys/block/sd[a-b] }
```

关闭指定设备的“状态 LED”和“故障 LED”：

```
> sudo ledctl off={ /dev/sda /dev/sdb }
```

要找到三个块设备，请运行以下命令之一（两个命令是等效的）：

```
> sudo ledctl locate=/dev/sda,/dev/sdb,/dev/sdc
> sudo ledctl locate={ /dev/sda /dev/sdb /dev/sdc }
```

12.3 更多信息

有关 LED 模式以及监视工具的详细信息，请参见以下资源：

- GitHub.com 上的 LEDMON 开源项目 (https://github.com/intel/ledmon.git)
- SGPIO 规范 SFF-8485 (https://ftp.seagate.com/sff/SFF-8485.PDF)
- IBPI 规范 SFF-8489 (https://ftp.seagate.com/sff/SFF-8489.PDF)

13 软件 RAID 查错

查看 `/proc/mdstats` 文件以确定 RAID 分区是否受损。如果磁盘出现故障，请关闭 Linux 系统并用以同样方式分区的新硬盘替换出现问题的硬盘。然后重启动您的系统并输入命令 **`mdadm /dev/mdX --add /dev/sdX`**。将 X 替换为您的特定设备标识符。这会自动将硬盘整合到 RAID 系统中并完全重新构造（适用于除 RAID 0 以外的所有 RAID 级别）。

尽管可以在重构建期间访问所有数据，但在 RAID 完全重构建之前，仍然可能遇到一些性能问题。

13.1 修复故障磁盘之后进行恢复

RAID 阵列中的磁盘可能会出于多种原因而发生故障。下面列出了最常见的原因：

- 磁盘媒体出现问题。
- 磁盘驱动器控制器发生故障。
- 与磁盘的连接断开。

在发生磁盘媒体或控制器故障时，需要更换或修复设备。如果未在 RAID 中配置热备用，则需要手动干预。

对于后一种情况，可以在修复连接（可能会自动修复）之后，使用 **`mdadm`** 命令自动重新添加发生故障的设备。

由于 **`md`**/**`mdadm`** 不能可靠地判断磁盘发生故障的原因，因此会臆测发生了严重的磁盘错误，并一直将任何发生操作失败的设备视为有故障，直到明确被告知该设备可靠为止。

在某些情况下（例如，储存设备包含内部 RAID 阵列），连接问题往往是设备发生故障的原因。在这种情况下，您可以告知 **`mdadm`**，在设备出现后，可以放心地使用 `--re-add` 自动重新添加该设备。为此，您可以将下面一行添加到 `/etc/mdadm.conf` 中：

```
POLICY action=re-add
```

请注意，仅当 udev 规则导致 **`mdadm -I`** `DISK_DEVICE_NAME` 在自发出现的任何设备上运行（默认行为），并且已配置 write-intent 位图（默认会配置）时，才会在设备重新出现之后自动重新添加该设备。

如果您希望此策略仅应用到某些设备而不应用到其余设备，可以将 `path=` 选项添加到 `/etc/mdadm.conf` 中的 `POLICY` 一行，以将非默认操作限制为只对选定的设备执行。可以使用通配符来识别设备组。有关详细信息，请参见 **`man 5 mdadm.conf`**。

IV 网络储存

14 iSNS for Linux 139

15 经由 IP 网络的大容量储存：iSCSI 146

16 以太网光纤通道储存：FCoE 172

17 NVMe over Fabric 182

18 管理设备的多路径 I/O 191

19 通过 NFS 共享文件系统 252

20 Samba 269

21 使用 Autofs 按需挂载 295

14 iSNS for Linux

储存区域网络 (SAN) 可包含分布在复杂网络间的许多磁盘驱动器。这会使设备发现和设备所有权变得复杂。iSCSI 发起端必须能够识别 SAN 中的储存资源，并确定是否可对其进行访问。

互联网储存名称服务 (iSNS) 是一项基于标准的服务，使用它可简化 TCP/IP 网络上 iSCSI 设备的自动发现、管理和配置。与光纤通道网络中的服务相比，iSNS 可提供智能的储存发现和管理服务。

如果没有 iSNS，您必须知道所需目标所在的每个节点的主机名或 IP 地址。此外，必须使用访问控制列表等机制，自行手动管理哪些发起端能够访问哪些目标。

重要：安全注意事项

由于网络流量未加密，只能在安全的内部网络环境中使用 iSNS。

14.1 iSNS 的工作原理

iSCSI 发起端要发现 iSCSI 目标，需要识别网络中的哪些设备是储存资源以及访问这些资源需要哪些 IP 地址。对 iSNS 服务器的查询会返回发起端有权访问的一个 iSCSI 目标和 IP 地址的列表。

通过使用 iSNS，您创建 iSNS 发现域，随后将 iSCSI 目标和发起端分组或组织到这些域中。通过将储存节点划分到域，您可以将每个主机的发现进程限制为在 iSNS 中注册的最合适的目标子集，这使得储存网络可通过降低不必要的发现数和限制每个主机在建立发现关系时花费的时间而按比例缩放。这使您可以控制和简化必须发现的目标和发起端的数目。

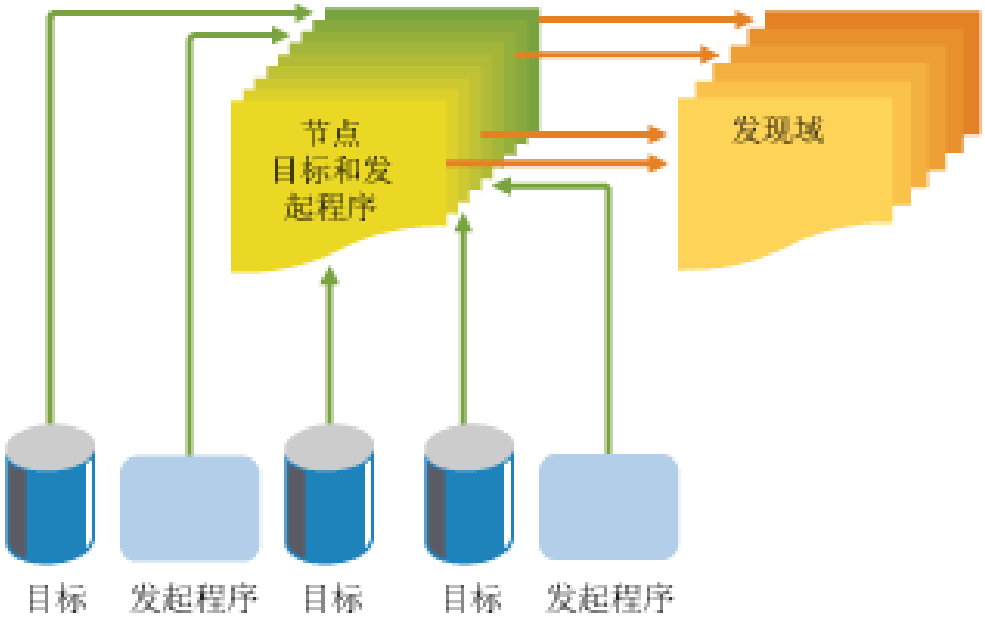


图 14.1： ISNS 发现域

iSCSI 目标和 iSCSI 发起端都可以使用 iSNS 客户端通过 iSNS 协议发起与 iSNS 服务器之间的事务。然后，它们在公共发现域中注册设备属性信息、下载有关其他注册客户端的信息，以及接收出现在其发现域中的事件的异步通知。

iSNS 服务器响应由 iSNS 客户端使用 iSNS 协议作出的 iSNS 协议查询和请求。iSNS 服务器启动 iSNS 协议状态更改通知，并正确储存由 iSNS 数据库中的注册请求提交的身份验证信息。

适用于 Linux 的 iSNS 提供的优点包括：

- 提供信息设备以供注册、发现和管理联网储存资产。
- 与 DNS 基础结构集成。
- 统一 iSCSI 储存的注册、发现和管理。
- 简化储存管理实施。
- 与其他发现方法相比，改进了可伸缩性。

iSNS 具备多项重要优势。

例如，在包含 100 个 iSCSI 发起端和 100 个 iSCSI 目标的设置中，所有 iSCSI 发起端都可尝试发现和连接 100 个 iSCSI 目标中的任何一个。通过将发起端和目标分组到发现域，您可以阻止某个部门的 iSCSI 发起端发现其他部门的 iSCSI 目标。

使用 iSNS 的另一项优势在于，iSCSI 客户端只需知道 iSNS 服务器的主机名或 IP 地址，而不必知道全部 100 个服务器的主机名或 IP 地址。

14.2 安装 iSNS Server for Linux

iSNS Server for Linux 随附于 SUSE Linux Enterprise Server，但默认不会对其加以安装或设置。您需要安装软件包 `open-isns` 并设置 iSNS 服务。

注意：iSNS 和 iSCSI 在同一服务器上

可以在安装了 iSCSI 目标或 iSCSI 发起端软件的同一服务器上安装 iSNS。不支持在同一服务器上同时安装 iSCSI 目标软件和 iSCSI 发起端软件。

安装 iSNS for Linux：

1. 启动 YaST 并选择网络服务 › iSNS 服务器。
2. 如果尚未安装 `open-isns`，则系统现在会提示您进行安装。通过单击安装确认安装。
3. “iSNS 服务”配置对话框会自动打开并显示服务选项卡。

4. 在启动服务中，选择以下选项之一：

   - **引导时：** iSNS 服务在服务器启动时自动启动。
   - **手动（默认）：** 必须在用于安装 iSNS 服务的服务器的控制台中手动输入 `sudo systemctl start isnsd` 来启动 iSNS 服务。

5. 指定以下防火墙设置：

   - **在防火墙上打开端口：** 选中该复选框以打开防火墙，并允许从远程计算机访问该服务。默认情况下，防火墙端口是关闭的。
   - **防火墙细节：** 如果打开防火墙端口，默认情况下会在所有网络接口上打开该端口。单击防火墙细节以选择要打开该端口的接口，选择要使用的网络接口，然后单击确定。

6. 单击确定可应用配置设置并完成安装。

7. 继续执行第 14.3 节 “配置 iSNS 发现域”。

14.3 配置 iSNS 发现域

要让 iSCSI 发起端和目标使用 iSNS 服务，它们必须属于某个发现域。

重要：iSNS 服务必须处于活动状态

必须已安装并运行 iSNS 服务，才能设置 iSNS 发现域。有关信息，请参见第 14.4 节 “启动 iSNS 服务”。

14.3.1 创建 iSNS 发现域

安装 iSNS 服务时，将会自动创建一个名为默认 DD 的默认发现域。已配置为使用 iSNS 的现有 iSCSI 目标和发起端会自动添加到默认的发现域。

创建新的发现域：

1. 启动 YaST，然后在网络服务下选择 iSNS 服务器。

2. 单击发现域选项卡。

   发现域区域列出了所有现有的发现域。您可以创建发现域，或删除现有的发现域。请注意，从域成员资格中删除某个 iSCSI 节点只是将其从域中去除，而不会删除该 iSCSI 节点。

   发现域成员区域列出指派给所选发现域的所有 iSCSI 节点。选择不同的发现域会使用该发现域的成员刷新该列表。您可以在所选发现域中添加和删除 iSCSI 节点。删除 iSCSI 节点会从域中删除该节点，但不会删除该 iSCSI 节点。

   创建 iSCSI 节点成员允许将尚未注册的节点添加为发现域成员。当 iSCSI 发起端或目标注册该节点时，它会成为该域的一部分。

   当 iSCSI 发起端执行发现请求时，iSNS 服务会返回作为同一发现域成员的所有 iSCSI 节点目标。

3. 单击创建发现域按钮。

   还可以选择现有发现域，然后单击删除按钮删除该发现域。

4. 指定创建的发现域的名称，然后单击确定。

5. 继续执行第 14.3.2 节 “向发现域添加 iSCSI 节点”。

14.3.2 向发现域添加 iSCSI 节点

1. 启动 YaST，然后在网络服务下选择 iSNS 服务器。

2. 单击 iSCSI 节点选项卡。

3. 检查节点列表以确保列出了要使用 iSNS 服务的 iSCSI 目标和发起端。
   如果未列出 iSCSI 目标或发起端，可能需要在节点上重启动 iSCSI 服务。为此，您可以运行

```
> sudo systemctl restart iscsid.socket
> sudo systemctl restart iscsi
```

   重启动发起端，或运行

```
> sudo systemctl restart target-isns
```

   重启动目标。
   可以选择某 iSCSI 节点，然后单击删除按钮从 iSNS 数据库中删除该节点。如果不再使用某 iSCSI 节点或已对其重命名，这十分有用。

除非删除或注释掉 iSCSI 配置文件的 iSNS 部分，否则，在重启动 iSCSI 服务或重引导服务器时，iSCSI 节点将再次自动添加到列表（iSNS 数据库）。

4. 单击发现域选项卡并选择所需的发现域。
5. 单击添加现有 iSCSI 节点，选择要添加到域中的节点，然后单击添加节点。
6. 为添加至发现域的节点重复上一步，当您完成添加节点时，再单击完成。
   请注意，一个 iSCSI 节点可属于多个发现域。

14.4 启动 iSNS 服务

必须在安装 iSNS 的服务器上启动它。如果您未将它设置为在引导时启动（有关细节，请参见第 14.2 节 “安装 iSNS Server for Linux”），请在终端控制台中输入以下命令：

```
> sudo systemctl start isnsd
```

您还可以对 iSNS 使用 **stop**、**status** 和 **restart** 选项。

14.5 更多信息

以下项目提供了有关 iSNS 和 iSCSI 的其他信息：

- 适用于 Linux 项目的 iSNS 服务器和客户端 (https://github.com/open-iscsi/open-isns)
- 适用于 Linux LIO iSCSI 目标的 iSNS 客户端 (https://github.com/open-iscsi/target-isns)
- 适用于 Linux 的 iSCSI 工具 (https://www.open-iscsi.com)

有关 iSNS 的一般信息，请参见 RFC 4171: Internet Storage Name Service（RFC 4171：互联网储存设备名称服务，网址：https://datatracker.ietf.org/doc/html/rfc4171）。

15 经由 IP 网络的大容量储存：iSCSI

提供充足的磁盘容量是计算机中心或支持服务器的任何站点的主要任务之一。通常为此目的使用光纤通道。iSCSI（互联网 SCSI）解决方案提供了光纤通道的低成本备用方案，可以充分利用商品服务器和以太网联网设备。Linux iSCSI 提供 iSCSI 发起端和 iSCSI LIO 目标软件，用于将 Linux 服务器连接到中心储存系统。

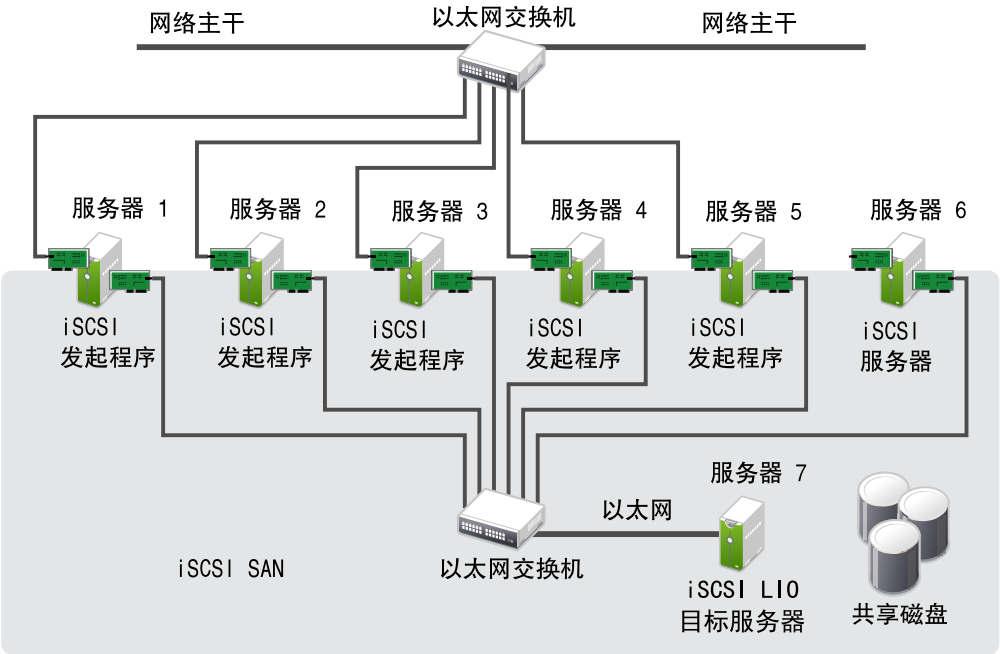


图 15.1： 使用 ISNS 服务器的 ISCSI SAN

注意：LIO

LIO (http://linux-iscsi.org) 是 Linux 适用的标准开源多协议 SCSI 目标。LIO 以 Linux 内核 2.6.38 及更高版本替代 STGT（SCSI 目标）框架成为 Linux 中的标准统一储存目标。在 SUSE Linux Enterprise Server 12 中，iSCSI LIO 目标服务器替换了先前版本中的 iSCSI 目标服务器。

iSCSI 是一种储存联网协议，可简化在块储存设备和服务器之间通过 TCP/IP 网络进行的 SCSI 包数据传输。iSCSI 目标软件在目标服务器上运行，并将逻辑单元定义为 iSCSI 目标设备。iSCSI 发起端软件在不同服务器上运行并连接到目标设备，以使此服务器上的储存设备可用。

iSCSI LIO 目标服务器与 iSCSI 发起端服务器之间通过在 LAN 的 IP 级别上发送 SCSI 包来进行通讯。当在发起端服务器上运行的应用程序启动针对 iSCSI LIO 目标设备的查询时，操作系统会生成必要的 SCSI 命令。然后，SCSI 命令会嵌入到 IP 包中并在需要时由软件进行加密，此软件通常为 **iSCSI 发起端**。包经由内部 IP 网络传输到相应的 iSCSI 远程站，该站称为 **iSCSI LIO 目标服务器**（简称为 **iSCSI 目标**）。

许多储存解决方案允许通过 iSCSI 进行访问，但是也可以运行提供 iSCSI 目标的 Linux 服务器。在此情况下，对已针对文件系统服务优化过的 Linux 服务器进行设置是很重要的。有关 RAID 的更多信息，请参见第 7 章 “软件 RAID 配置”。

15.1 安装 iSCSI LIO 目标服务器和 iSCSI 发起端

系统会默认安装 iSCSI 发起端（软件包 `open-iscsi` 和 `yast2-iscsi-client`），iSCSI LIO 目标软件包则需要手动安装。

重要：发起端和目标位于同一服务器

尽管可以在同一系统中运行发起端和目标，但不建议采用此设置。

要安装 iSCSI LIO 目标服务器，请在终端控制台中运行以下命令：

```
> sudo zypper in yast2-iscsi-lio-server
```

如果您需要安装 iSCSI 发起端或任何依赖项，请运行命令 **`sudo zypper in yast2-iscsi-client`**。

也可以使用 YaST 软件管理模块来进行安装。

除了上述软件包之外，任何额外需要的包将由安装程序自动提取，或者在您第一次运行相应的 YaST 模块时安装。

15.2 设置 iSCSI LIO 目标服务器

本章说明如何使用 YaST 配置 iSCSI LIO 目标服务器以及如何设置 iSCSI LIO 目标设备。您可以使用任何 iSCSI 发起端访问目标设备。

15.2.1 iSCSI LIO 目标服务启动和防火墙设置

iSCSI LIO 目标服务默认配置为手动启动。您可以将该服务配置为在引导时自动启动。如果服务器上使用防火墙，并且您想让 iSCSI LIO 目标对其他计算机可用，则必须为想要用于目标访问的每一个适配器打开防火墙中的端口。TCP 端口 3260 是 IANA（互联网号码分配机构）所定义的 iSCSI 协议的端口号。

1. 启动 YaST，然后启动网络服务 › iSCSI LIO 目标。

2. 切换到服务选项卡。

3. 在服务启动下，指定 iSCSI LIO 目标服务的启动方式：

   - **引导时：** 服务器重启动时自动启动服务。
   - **手动：**（默认）服务器重启动之后，您必须运行 `sudo systemctl start targetcli` 命令来手动启动该服务。只有启动服务后才可以使用目标设备。

4. 如果在服务器上使用防火墙，并且希望 iSCSI LIO 目标对其他计算机可用，请为要用于目标访问的每个适配器接口打开防火墙中的端口 3260。如果端口对于所有网络接口均为关闭，则其他计算机将无法使用 iSCSI LIO 目标。
   如果您未在服务器上使用防火墙，则防火墙设置会被禁用。在此情况下，请跳过下面的步骤，并单击完成离开配置对话框，或切换到另一个选项卡继续配置。

a. 在服务选项卡上，选中打开防火墙中的端口复选框，以启用防火墙设置。

b. 单击防火墙细节，查看或配置要使用的网络接口。列出所有可用的网络接口，并且默认全部选中。取消选择**不**应打开的端口上的所有接口。单击确定保存您的设置。

5. 单击完成，以保存和应用 iSCSI LIO 目标服务设置。

15.2.2 配置身份验证以发现 iSCSI LIO 目标和发起端

iSCSI LIO 目标服务器软件支持 PPP-CHAP（点对点协议-质询握手身份验证协议），该协议是互联网工程任务组 (IETF) RFC 1994 (https://datatracker.ietf.org/doc/html/rfc1994) 中定义的一种三向身份验证方法。服务器使用此身份验证方法来发现 iSCSI LIO 目标与发起端，并不用于访问目标上的文件。如果不希望限制对发现的访问，则使用无身份验证。不进行发现身份验证选项默认处于启用状态。不需要经过身份验证，此服务器上的所有 iSCSI LIO 目标都可由同一网络上的任何 iSCSI 发起端发现。

如果是为了更为安全的配置而需要身份验证，则可以使用传入身份验证、传出身份验证，或二者皆使用。按发起端进行身份验证要求 iSCSI 发起端证明它有权针对 iSCSI LIO 目标运行发现。发起端必须提供传入用户名和口令。按目标进行身份验证要求 iSCSI LIO 目标向发起端证明它是预期的目标。iSCSI LIO 目标必须向 iSCSI 发起端提供传出用户名和口令。用于传入和传出发现的口令必须不同。如果启用发现身份验证，则其设置将应用到所有 iSCSI LIO 目标组。

重要：安全性

为安全起见，建议您对生产环境中的目标与发起端发现使用身份验证。

配置 iSCSI LIO 目标的身份验证自选设置：

1. 启动 YaST，然后启动网络服务 › iSCSI LIO 目标。

2. 切换到全局选项卡。

3. 身份验证默认处于禁用状态（不进行发现身份验证）。要启用身份验证，请选择按发起端进行身份验证和/或传出身份验证。
4. 提供所选身份验证方法的身份凭证。传入和传出发现的用户名和口令对必须不同。
5. 单击完成保存并应用设置。

15.2.3 准备储存空间

在为 iSCSI 目标服务器配置 LUN 之前，必须准备好要使用的储存空间。可以将整个未格式化的块设备用作单个 LUN，也可以将一台设备划分为数个未格式化的分区，并将每个分区用作单独的 LUN。iSCSI 目标配置会将 LUN 导出到 iSCSI 发起端。

您可以使用 YaST 中的分区程序或命令行来设置分区。有关细节，请参考《部署指南》, 第 10 章 “专家分区程序”, 第 10.1 节 “使用专家分区程序”。iSCSI LIO 目标可将未格式化的分区用于 Linux、Linux LVM 或 Linux RAID 文件系统 ID。

重要：不要挂载 iSCSI 目标设备

设置完用作 iSCSI 目标的设备或分区后，切勿通过其本地路径直接对其进行访问。不要在目标服务器上挂载分区。

15.2.3.1 对虚拟环境中的设备进行分区

您可以将虚拟机 guest 服务器作为 iSCSI LIO 目标服务器使用。本节说明如何给 Xen 虚拟机指派分区。您还可以使用 SUSE Linux Enterprise Server 支持的其他虚拟环境。

在 Xen 虚拟环境中，必须将要用于 iSCSI LIO 目标设备的储存空间指派给 guest 虚拟机，然后作为 guest 环境中的虚拟磁盘访问该空间。每个虚拟磁盘都可以是一个物理块设备，如整个磁盘、分区或卷，或者可以是一个基于文件的磁盘映像，其中虚拟磁盘是 Xen 主机服务器上较大物理磁盘上的单个映像文件。为了获得最佳性能，请从物理磁盘或分区创建每个虚拟磁盘。为 guest 虚拟机设置完虚拟磁盘后，请启动 guest 服务器，然后按照物理服务器所用的同一过程将新的空虚拟磁盘配置为 iSCSI 目标设备。

在 Xen 主机服务器上创建基于文件的磁盘映像，然后将其指派给 Xen guest 服务器。默认情况下，Xen 将基于文件的磁盘映像储存在 `/var/lib/xen/images/VM_NAME` 目录中，其中 `VM_NAME` 是虚拟机的名称。

15.2.4 设置 iSCSI LIO 目标组

您可以使用 YaST 配置 iSCSI LIO 目标设备。YaST 使用 **`targetcli`** 软件。iSCSI LIO 目标可使用采用 Linux、Linux LVM 或 Linux RAID 文件系统 ID 的分区。

重要：分区

在开始之前，请选择要用于后端储存的分区。不一定要格式化这些分区，iSCSI 客户端可以在连接分区时对这些分区进行格式化，并重写所有现有格式。

1. 启动 YaST，然后启动网络服务 › iSCSI LIO 目标。
2. 切换到目标选项卡。

3. 单击添加，然后定义新的 iSCSI LIO 目标组以及设备：
   iSCSI LIO 目标软件将自动填写 目标、标识符、门户组、IP 地址以及端口号字段。默认会选择使用身份验证。

   a. 如果您有多个网络接口，请使用 IP 地址下拉框选择要用于此目标组的网络接口的 IP 地址。要使服务器在所有地址下都可访问，请选择绑定所有 IP 地址。

   b. 如果您不想对此目标组进行发起端身份验证，请取消选择使用身份验证（不建议此做法）。

   c. 单击添加。输入设备或分区的路径，或单击浏览添加该路径。也可以指定名称，然后单击确定。系统将自动生成 LUN 号（从 0 开始）。如果将该字段保留为空，则会自动生成名称。

   d. （可选）重复前面的步骤将目标添加到此目标组。

   e. 将所有所需的目标添加到该组后，单击下一步。

4. 在修改 iSCSI 目标发起端设置页面上，配置允许访问目标组中 LUN 的发起端的信息：

在为目标组指定了至少一个发起端后，编辑 LUN、编辑身份验证、删除和复制按钮才会启用。您可以使用添加或复制来为目标组添加发起端：

修改 ISCSI 目标：选项

- **添加：** 为选中的 iSCSI LIO 目标组添加新的发起端项。
- **编辑 LUN：** 配置将 iSCSI LIO 目标组中的哪些 LUN 映射到选定发起端。您可以将分配的每一个目标都映射到某个首选的发起端。
- **编辑身份验证：** 为选定发起端配置首选身份验证方法。您可以指定无身份验证，也可以配置传入身份验证、传出身份验证或二者皆配置。
- **删除：** 将选定发起端项从分配给目标组的发起端列表中去除。
- **复制：** 添加具有与选定发起端项相同 LUN 映射和身份验证设置的新发起端项。如此，您便可以轻松地将相同的共享 LUN 逐一分配给群集中的每一个节点。

a. 单击添加，指定发起端名称，选中或取消选中从 TPG 中导入 LUN 复选框，然后单击确定保存设置。

b. 选择发起端项，单击编辑 LUN，修改 LUN 映射，以指定将 iSCSI LIO 目标组中的哪些 LUN 分配给选定发起端，然后单击确定保存更改。

如果 iSCSI LIO 目标组由多个 LUN 组成，您可以将一个或多个 LUN 分配给选定发起端。默认情况下，会将该组中的每个可用 LUN 都指派给某个发起端 LUN。

要修改 LUN 分配，请完成以下一项或多项操作：

- **添加：** 单击添加创建新的发起端 LUN 项，然后使用更改下拉框将一个目标 LUN 映射到该项。
- **删除：** 选择发起端 LUN 项，然后单击删除去除目标 LUN 映射。
- **更改：** 选择发起端 LUN 项，然后使用更改下拉框选择要与之映射的目标 LUN。

典型的分配计划包括：

- 单个服务器列为一个发起端。将目标组中的所有 LUN 分配给它。
  您可以使用此分组策略，为指定服务器逻辑分组 iSCSI SAN 储存。
- 多个独立服务器列为多个发起端。将一个或多个目标 LUN 分配给每一个服务器。将每个 LUN 只分配给一个服务器。
  您可以使用此分组策略，为数据中心中的指定部门或服务类别逻辑分组 iSCSI SAN 储存。
- 群集的每个节点列为一个发起端。将所有共享的目标 LUN 分配给每一个节点。所有节点都将连接到设备，但对于大多数文件系统，群集软件会锁定设备不让它们访问，并且一次只在一个节点上挂载该设备。共享的文件系统（例如 OCFS2）可让多个节点同时挂载相同的文件结构，并以读写访问权打开相同文件。
  您可以使用此分组策略，为指定服务器群集逻辑分组 iSCSI SAN 储存。

c. 选择发起端项，单击编辑身份验证，指定发起端的身份验证设置，然后单击确定保存设置。

您可以要求不进行发现身份验证，也可以配置按发起端进行身份验证和/或传出身份验证。您仅可为每个发起端指定一对用户名和口令。对于发起端的传入和传出身份验证，身份凭证可以有所不同。每个发起端的身份凭证均可不同。

d. 对每个可以访问此目标组的 iSCSI 发起端重复上述步骤。

e. 配置完发起端指派后，单击下一步。

5. 单击完成保存并应用设置。

15.2.5 修改 iSCSI LIO 目标组

您可以按照如下方式修改现有 iSCSI LIO 目标组：

- 在目标组中添加或删除目标 LUN 设备
- 为目标组添加或去除发起端
- 为目标组的发起端修改发起端 LUN 到目标 LUN 的映射
- 修改发起端身份验证（传入、传出或二者）的用户名和口令身份凭证。

查看或修改 iSCSI LIO 目标组的设置：

1. 启动 YaST，然后启动网络服务 › iSCSI LIO 目标。
2. 切换到目标选项卡。
3. 选择要修改的 iSCSI LIO 目标组，然后单击编辑。
4. 在“修改 iSCSI 目标 LUN 设置”页面上，将 LUN 添加到目标组，编辑 LUN 指派或从组中删除目标 LUN。在对组完成所有想要的更改后，单击下一步。
   有关选项信息，请参见修改 iSCSI 目标：选项。
5. 在“修改 iSCSI 目标发起端设置”页面上，配置允许访问目标组中 LUN 的发起端的信息。在对组完成所有想要的更改后，单击下一步。
6. 单击完成保存并应用设置。

15.2.6 删除 iSCSI LIO 目标组

删除 iSCSI LIO 目标组会去除组的定义以及发起端的相关设置，包括 LUN 映射和身份验证凭证。该操作不会销毁分区上的数据。要再次赋予发起端访问权限，您可以将目标 LUN 分配给不同的或新的目标组，并为其配置发起端访问权限。

1. 启动 YaST，然后启动网络服务 › iSCSI LIO 目标。
2. 切换到目标选项卡。
3. 选择要删除的 iSCSI LIO 目标组，然后单击删除。
4. 系统提示时，单击继续确认删除，或单击取消予以取消。
5. 单击完成保存并应用设置。

15.3 配置 iSCSI 发起端

iSCSI 发起端可用于连接到任意 iSCSI 目标。连接不限于第 15.2 节 “设置 iSCSI LIO 目标服务器” 说明的 iSCSI 目标解决方案。iSCSI 发起端配置包括两个主要步骤 — 发现可用 iSCSI 目标和设置 iSCSI 会话。这两个步骤都可通过 YaST 完成。

15.3.1 使用 YaST 配置 iSCSI 发起端

在 YaST 中，“iSCSI 发起端概述” 包括三个选项卡：

服务：

可使用服务选项卡来在引导时启用 iSCSI 发起端。它还允许设置唯一的发起端名称和 iSNS 服务器以用于发现。

已连接目标：

已连接目标选项卡概述了当前已连接的 iSCSI 目标。与已发现目标选项卡类似，它还提供用于向系统添加新目标的选项。

已发现目标：

已发现目标选项卡使您能够手动发现网络中的 iSCSI 目标。

15.3.1.1 配置 iSCSI 发起端

1. 启动 YaST，然后启动网络服务 › iSCSI 发起端。
2. 切换到服务选项卡。

3. 在写入配置后下，定义在发生配置更改时要执行什么操作。请记住，可用选项取决于服务的当前状态。
   保持当前状态选项会使服务保持相同状态。

4. 在重启后菜单中指定重启后要执行的操作：

   - 引导时启动 - 引导时自动启动服务。
   - 按需启动 - 将运行关联的套接字，并根据需要启动服务。
   - 不启动 - 服务不自动启动。
   - 保留当前设置 - 不更改服务配置。

5. 指定或校验发起端名称。
   为此服务器上的 iSCSI 发起端指定格式正确的 iSCSI 限定名称 (IQN)。此发起端名称在网络上必须具有全局唯一性。IQN 使用以下常规格式：

```
iqn.yyyy-mm.com.mycompany:n1:n2
```

   其中，n1 和 n2 是字母数字字符。例如：

```
iqn.1996-04.de.suse:01:a5dfcea717a
```

发起端名称将用服务器上文件 `/etc/iscsi/initiatorname.iscsi` 中的相应值自动填充。

如果服务器提供 iBFT (iSCSI Boot Firmware Table) 支持，则发起端名称将用 IBFT 中的相应值填充，并且您不能在此界面中更改发起端名称。不过，您可以使用 BIOS 设置来进行修改。iBFT 是指包含各种对 iSCSI 引导进程有用的参数的信息块，包括针对服务器的 iSCSI 目标与发起端描述。

6. 使用以下方法之一发现网络上的 iSCSI 目标。

   - **iSNS：** 要使用 iSNS (Internet Storage Name Service) 来发现 iSCSI 目标，请按第 15.3.1.2 节 “使用 iSNS 发现 iSCSI 目标” 中所述继续操作。
   - **已发现目标：** 要手动发现 iSCSI 目标设备，请按第 15.3.1.3 节 “手动发现 iSCSI 目标” 中所述继续操作。

15.3.1.2 使用 iSNS 发现 iSCSI 目标

必须在环境中已安装并配置 iSNS 服务器后，才能使用此选项。有关信息，请参见第 14 章 “iSNS for Linux” 。

1. 在 YaST 中，选择 iSCSI 发起端，然后选择服务选项卡。
2. 指定 iSNS 服务器的 IP 地址和端口。默认端口为 3205。
3. 单击确定保存并应用更改。

15.3.1.3 手动发现 iSCSI 目标

对要从已设置 iSCSI 发起端的服务器访问的每个 iSCSI 目标服务器重复以下过程。

1. 在 YaST 中，选择 iSCSI 发起端，然后选择已发现目标选项卡。
2. 单击发现打开 iSCSI 发起端发现对话框。
3. 输入 IP 地址并根据需要更改端口。默认端口为 3260。
4. 如果要求进行身份验证，请取消选择不进行发现身份验证，然后为按发起端进行身份验证或按目标进行身份验证指定身份凭证。

5. 单击下一步启动发现并连接到 iSCSI 目标服务器。
6. 如果要求提供身份凭证，在成功发现后，请使用连接激活目标。
   系统会提示您输入身份验证凭证以使用所选 iSCSI 目标。
7. 单击下一步完成配置。
   该目标会立即出现在已连接目标中，现在便可使用虚拟 iSCSI 设备了。
8. 单击确定保存并应用更改。
9. 您可以使用 **`lsscsi`** 命令查找 iSCSI 目标设备的本地设备路径。

15.3.1.4 为 iSCSI 目标设备设置启动首选项

1. 在 YaST 中，选择 iSCSI 发起端，然后选择已连接目标选项卡以查看当前连接到服务器的 iSCSI 目标设备的列表。
2. 选择要管理的 iSCSI 目标设备。
3. 单击切换启动修改设置：

   自动： 此选项用于 iSCSI 服务本身启动时要连接的 iSCSI 目标。这是典型配置。

   引导时： 此选项用于引导期间要连接的 iSCSI 目标；即，当根目录 (/) 位于 iSCSI 上时。这样，iSCSI 目标设备在服务器引导时将从 initrd 进行评估。此选项在无法从 iSCSI 引导的平台（例如 IBM Z）上会被忽略。因此，在这些平台上不应使用该选项，而应使用自动。
4. 单击确定保存并应用更改。

15.3.2 手动设置 iSCSI 发起端

发现和配置 iSCSI 连接都要求 iscsid 正在运行。首次运行发现时，将在 `/etc/iscsi/` 目录中创建 iSCSI 发起端的内部数据库。

如果发现受口令保护，则向 iscsid 提供身份验证信息。由于首次执行发现时内部数据库并不存在，因此此时无法使用内部数据库。相反，必须编辑配置文件 `/etc/iscsid.conf` 以提供信息。要为发现添加口令信息，请在 `/etc/iscsid.conf` 末尾添加以下行：

```
discovery.sendtargets.auth.authmethod = CHAP
discovery.sendtargets.auth.username = USERNAME
discovery.sendtargets.auth.password = PASSWORD
```

发现会将所有接收到的值储存在一个内部持久数据库中。此外，它会显示所有检测到的目标。使用以下命令运行此发现：

```
> sudo iscsiadm -m discovery --type=st --portal=TARGET_IP
```

输出如下所示：

```
10.44.171.99:3260,1 iqn.2006-02.com.example.iserv:systems
```

要发现 iSNS 服务器上的可用目标，请使用以下命令：

```
sudo iscsiadm --mode discovery --type isns --portal TARGET_IP
```

对于 iSCSI 目标上定义的每个目标，将显示一行。有关储存数据的更多信息，请参见第 15.3.3 节 “iSCSI 发起端数据库”。

`iscsiadm` 的特殊 **`--login`** 选项会创建所有需要的设备：

```
> sudo iscsiadm -m node -n iqn.2006-02.com.example.iserv:systems --login
```

新生成的设备会显示在 **`lsscsi`** 的输出中，现在便可挂载该设备。

15.3.3 iSCSI 发起端数据库

iSCSI 发起端发现的所有信息都储存在位于 `/etc/iscsi` 中的两个数据库文件中。一个数据库用于发现的目标，另一个数据库用于发现的节点。访问数据库时，首先必须选择是希望从发现获取数据还是从节点数据库获取数据。可使用 **`iscsiadm`** 的参数 `-m discovery` 和 `-m node` 来执行此操作。使用 **`iscsiadm`** 搭配其中一个参数，可提供储存记录的概述：

```
> sudo iscsiadm -m discovery
10.44.171.99:3260,1 iqn.2006-02.com.example.iserv:systems
```

本示例中的目标名称为 `iqn.2006-02.com.example.iserv:systems`。与此特殊数据集相关的所有操作都需要此名称。要检查 ID 为 `iqn.2006-02.com.example.iserv:systems` 数据记录的内容，可使用以下命令：

```
> sudo iscsiadm -m node --targetname iqn.2006-02.com.example.iserv:systems
node.name = iqn.2006-02.com.example.iserv:systems
node.transport_name = tcp
node.tpgt = 1
node.active_conn = 1
node.startup = manual
node.session.initial_cmdsn = 0
node.session.reopen_max = 32
node.session.auth.authmethod = CHAP
node.session.auth.username = joe
node.session.auth.password = ********
node.session.auth.username_in = EMPTY
node.session.auth.password_in = EMPTY
node.session.timeo.replacement_timeout = 0
node.session.err_timeo.abort_timeout = 10
node.session.err_timeo.reset_timeout = 30
node.session.iscsi.InitialR2T = No
node.session.iscsi.ImmediateData = Yes
....
```

要编辑这些变量的值，可将 **`iscsiadm`** 与更新操作一起使用。例如，如果希望 iscsid 在初始化时登录 iSCSI 目标，则将变量 `node.startup` 的值设置为 `automatic`：

```
sudo iscsiadm -m node -n iqn.2006-02.com.example.iserv:systems \
-p ip:port --op=update --name=node.startup --value=automatic
```

使用 `delete` 操作移除过时的数据集。如果目标 `iqn.2006-02.com.example.iserv:systems` 不再是有效的记录，请使用以下命令删除此记录：

```
> sudo iscsiadm -m node -n iqn.2006-02.com.example.iserv:systems \
-p ip:port --op=delete
```

重要：无确认

使用此选项时应谨慎，因为它会删除该记录而无任何附加确认提示。

要获取所有已发现目标的列表，请运行 **`sudo iscsiadm -m node`** 命令。

15.4 使用 targetcli-fb 设置软件目标

`targetcli` 是用于管理 LinuxIO (LIO) 目标子系统配置的外壳。您可以交互方式调用该外壳，也可以像在传统的外壳中那样每次执行一条命令。类似于传统外壳，您可以使用 **`cd`** 命令遍历 targetcli 功能层次结构，并使用 **`ls`** 命令列出内容。

可用的命令取决于当前目录。虽然每个目录都有其各自的命令集，但也有一些命令可在所有目录中使用（例如 **`cd`** 和 **`ls`** 命令）。

`targetcli` 命令的格式如下：

```
[DIRECTORY] command [ARGUMENTS]
```

您可以在任何目录中使用 **`help`** 命令来查看可用命令列表，或查看有关命令的特定信息。

`targetcli` 工具是 `targetcli-fb` 软件包的一部分。此软件包已在官方 SUSE Linux Enterprise Server 软件源中提供，可使用以下命令进行安装：

```
> sudo zypper install targetcli-fb
```

安装 `targetcli-fb` 软件包后，启用 `targetcli` 服务：

```
> sudo systemctl enable targetcli
> sudo systemctl start targetcli
```

要切换到 targetcli 外壳，请以 root 身份运行 **`targetcli`**：

```
> sudo targetcli
```

然后，可以运行 **`ls`** 命令来查看默认配置。

```
/> ls
o- / ............................ [...]
  o- backstores ................. [...]
  | o- block ..... [Storage Objects: 0]
  | o- fileio .... [Storage Objects: 0]
  | o- pscsi ..... [Storage Objects: 0]
  | o- ramdisk ... [Storage Objects: 0]
  | o- rbd ....... [Storage Objects: 0]
  o- iscsi ............... [Targets: 0]
  o- loopback ............ [Targets: 0]
```

```
  o- vhost ............... [Targets: 0]
  o- xen-pvscsi .......... [Targets: 0]
/>
```

如 **ls** 命令的输出中所示，尚未配置任何后端。因此，第一步是配置一个受支持软件目标。

targetcli 支持以下后端：

- `fileio`：本地映像文件
- `block`：专用磁盘或分区上的块储存
- `pscsi`：SCSI 直通设备
- `ramdisk`：基于内存的后端
- `rbd`：Ceph RADOS 块设备

为了熟悉 targetcli 的功能，请使用 **create** 命令将本地映像文件设置为软件目标：

```
/backstores/fileio create test-disc /alt/test.img 1G
```

这会在指定的位置（在本例中为 `/alt`）创建 1 GB 的 `test.img` 映像。运行 **ls**，此时应会看到以下结果：

```
/> ls
o- / ......................................................... [...]
  o- backstores .............................................. [...]
  | o- block ................................... [Storage Objects: 0]
  | o- fileio .................................. [Storage Objects: 1]
  | | o- test-disc ... [/alt/test.img (1.0GiB) write-back deactivated]
  | |   o- alua ......     ......................... [ALUA Groups: 1]
  | |     o- default_tg_pt_gp      .... [ALUA state: Active/optimized]
  | o- pscsi ................................... [Storage Objects: 0]
  | o- ramdisk ................................. [Storage Objects: 0]
  | o- rbd ..................................... [Storage Objects: 0]
  o- iscsi ........................................... [Targets: 0]
  o- loopback ........................................ [Targets: 0]
  o- vhost ........................................... [Targets: 0]
  o- xen-pvscsi ...................................... [Targets: 0]
/>
```

输出中指出，已在 `/backstores/fileio` 目录下创建一个名为 `test-disc` 的基于文件的备用存储，并已链接到创建的文件 `/alt/test.img`。请注意，新的备用存储尚未激活。

下一步是将一个 iSCSI 目标前端连接到该后端储存。每个目标都必须有一个 `IQN`（iSCSI 限定的名称）。最常用的 IQN 格式如下：

```
iqn.YYYY-MM.NAMING-AUTHORITY:UNIQUE-NAME
```

必须提供 IQN 的以下部分：

- `YYYY-MM`：建立命名机构的年份和月份
- `NAMING-AUTHORITY`：命名机构的互联网域名的反向语法
- `UNIQUE-NAME`：命名机构选择的唯一域名

例如，对于域 `open-iscsi.com`，IQN 可以是：

```
iqn.2005-03.com.open-iscsi:UNIQUE-NAME
```

创建 iSCSI 目标时，**`targetcli`** 命令允许您指派自己的 IQN，只要该 IQN 遵循指定的格式即可。您还可以在创建目标时省略名称，让该命令为您创建 IQN，例如：

```
/> iscsi/ create
```

再次运行 **`ls`** 命令：

```
/> ls
o- / ..................................................................... [...]
  o- backstores ........................................................ [...]
  | o- block ........................................ [Storage Objects: 0]
  | o- fileio ....................................... [Storage Objects: 1]
  | | o- test-disc ....... [/alt/test.img (1.0GiB) write-back deactivated]
  | |   o- alua ........................................... [ALUA Groups: 1]
  | |     o- default_tg_pt_gp ............. [ALUA state: Active/optimized]
  | o- pscsi ......................................... [Storage Objects: 0]
  | o- ramdisk ....................................... [Storage Objects: 0]
  | o- rbd ........................................... [Storage Objects: 0]
  o- iscsi ...................................................... [Targets: 1]
  | o- iqn.2003-01.org.linux-iscsi.e83.x8664:sn.8b35d04dd456 ... [TPGs: 1]
  |   o- tpg1 ................................... [no-gen-acls, no-auth]
```

```
  |     o- acls ..................................................... [ACLs: 0]
  |     o- luns ..................................................... [LUNs: 0]
  |     o- portals ............................................ [Portals: 1]
  |       o- 0.0.0.0:3260 ........................................... [OK]
  o- loopback ...................................................... [Targets: 0]
  o- vhost ......................................................... [Targets: 0]
  o- xen-pvscsi .................................................... [Targets: 0]
/>
```

输出显示创建了 iSCSI 目标节点，并自动为其生成 IQN `iqn.2003-01.org.linux-iscsi.e83.x8664:sn.8b35d04dd456`

请注意，**`targetcli`** 还创建并启用了默认的目标门户组 `tpg1`。这是因为位于根级别的变量 `auto_add_default_portal` 和 `auto_enable_tpgt` 默认设置为 `true`。

该命令还使用 `0.0.0.0` IPv4 通配符创建了默认门户。这意味着，任何 IPv4 地址都可以访问配置的目标。

下一步是为 iSCSI 目标创建 LUN（逻辑单元号）。执行此操作的最佳做法是让 **`targetcli`** 自动指派其名称和编号。切换到 iSCSI 目标所在的目录，然后在 `lun` 目录中使用 **`create`** 命令为备用存储指派 LUN。

```
/> cd /iscsi/iqn.2003-01.org.linux-iscsi.e83.x8664:sn.8b35d04dd456/
/iscsi/iqn.2003-01.org.linux-iscsi.e83.x8664:sn.8b35d04dd456> cd tpg1
/iscsi/iqn.2003-01.org.linux-iscsi.e83.x8664:sn.8b35d04dd456/tpg1> luns/
create /backstores/fileio/test-disc
```

运行 **`ls`** 命令以查看更改：

```
/iscsi/iqn.2003-01.org.linux-iscsi.e83.x8664:sn.8b35d04dd456/tpg1> ls
o- tpg1 ............................................ [no-gen-acls, no-auth]
      o- acls ........................................................ [ACLs: 0]
      o- luns ........................................................ [LUNs: 1]
      | o- lun0 ....... [fileio/test-disc (/alt/test.img) (default_tg_pt_gp)]
      o- portals ............................................ [Portals: 1]
        o- 0.0.0.0:3260 ............................................... [OK]
```

现在，即创建了一个具有 1 GB 基于文件的备用存储的 iSCSI 目标。该目标的名称为 `iqn.2003-01.org.linux-iscsi.e83.x8664:sn.8b35d04dd456`，可从系统的任何网络端口访问它。

最后，需要确保发起端能够访问配置的目标。要实现此目的，一种方法是为每个发起端创建一个允许其连接到目标的 ACL 规则。在这种情况下，必须使用其 IQN 列出每个所需的发起端。可以在 `/etc/iscsi/initiatorname.iscsi` 文件中找到现有发起端的 IQN。使用以下命令添加所需的发起端（在本例中为 `iqn.1996-04.de.suse:01:54cab487975b`）：

```
/iscsi/iqn.2003-01.org.linux-iscsi.e83.x8664:sn.8b35d04dd456/tpg1> acls/ create
 iqn.1996-04.de.suse:01:54cab487975b
Created Node ACL for iqn.1996-04.de.suse:01:54cab487975b
Created mapped LUN 0.
/iscsi/iqn.2003-01.org.linux-iscsi.e83.x8664:sn.8b35d04dd456/tpg1>
```

您也可以在不限制访问的演示模式下运行目标。此方法的安全性较低，但适合演示目的，并可以在封闭的网络中运行。要启用演示模式，请使用以下命令：

```
/iscsi/iqn.2003-01.org.linux-iscsi.e83.x8664:sn.8b35d04dd456/tpg1> set attribute
 generate_node_acls=1
/iscsi/iqn.2003-01.org.linux-iscsi.e83.x8664:sn.8b35d04dd456/tpg1> set attribute
 demo_mode_write_protect=0
```

最后一步是使用根目录中可用的 **`saveconfig`** 命令保存创建的配置：

```
/> saveconfig /etc/target/example.json
```

如果在某个时间点您需要从保存的文件恢复配置，需要先清除当前配置。请记住，除非先保存配置，否则清除当前配置会导致数据丢失。使用以下命令清除并重新装载配置：

```
/> clearconfig
As a precaution, confirm=True needs to be set
/> clearconfig confirm=true
All configuration cleared
/> restoreconfig /etc/target/example.json
Configuration restored from /etc/target/example.json
/>
```

要测试配置的目标是否正常工作，请使用同一系统上安装的 `open-iscsi` iSCSI 发起端连接到该目标（请将 `HOSTNAME` 替换为本地计算机的主机名）：

```
> iscsiadm -m discovery -t st -p HOSTNAME
```

此命令将返回找到的目标列表，例如：

```
192.168.20.3:3260,1 iqn.2003-01.org.linux-iscsi.e83.x8664:sn.8b35d04dd456
```

然后，可以使用 **login** iSCSI 命令连接到列出的目标。这会使目标可用作本地磁盘。

15.5 安装时使用 iSCSI 磁盘

使用支持 iSCSI 的固件时，支持从 AMD64/Intel 64 和 IBM POWER 体系结构上的 iSCSI 磁盘引导。

要在安装过程中使用 iSCSI 磁盘，必须将以下参数添加到引导参数行：

```
withiscsi=1
```

安装过程中会额外显示一个屏幕，让您可以将 iSCSI 磁盘连接到系统，并在安装过程中加以使用。

注意：挂载点支持

引导期间，iSCSI 设备将异步显示。虽然 initrd 可确保为根文件系统正确设置这些设备，但对于任何其他文件系统或挂载点（例如 `/usr`），并无此类保证。因此，任何系统挂载点（例如 `/usr` 或 `/var`）都不受支持。要使用这些设备，请确保正确同步相应的服务和设备。

15.6 iSCSI 查错

本节介绍一些已知问题和针对 iSCSI 目标及 iSCSI 发起端问题的可行的解决方案。

15.6.1 在 iSCSI LIO 目标服务器上设置目标 LUN 时发生门户错误

添加或编辑 iSCSI LIO 目标组时发生错误：

```
Problem setting network portal IP_ADDRESS:3260
```

`/var/log/YasT2/y2log` 日志文件包含以下错误：

```
find: `/sys/kernel/config/target/iscsi': No such file or directory
```

若 iSCSI LIO 目标服务器软件当前未运行则会发生此问题。要解决此问题，请退出 YaST，并在命令行中使用 **systemctl start targetcli** 命令手动启动 iSCSI LIO，然后重试。

您也可以输入以下内容来检查 **configfs**、**iscsi_target_mod** 和 **target_core_mod** 是否载入。随即显示响应示例。

```
> sudo lsmod | grep iscsi
iscsi_target_mod      295015  0
target_core_mod       346745  4
iscsi_target_mod,target_core_pscsi,target_core_iblock,target_core_file
configfs               35817  3 iscsi_target_mod,target_core_mod
scsi_mod              231620  16
iscsi_target_mod,target_core_pscsi,target_core_mod,sg,sr_mod,mptctl,sd_mod,
scsi_dh_rdac,scsi_dh_emc,scsi_dh_alua,scsi_dh_hp_sw,scsi_dh,libata,mptspi,
mptscsih,scsi_transport_spi
```

15.6.2 iSCSI LIO 目标在其他计算机上不可见

如果目标服务器上使用防火墙，则必须打开要使用的 iSCSI 端口，以允许其他计算机看到 iSCSI LIO 目标。有关信息，请参见第 15.2.1 节 “iSCSI LIO 目标服务启动和防火墙设置”。

15.6.3 iSCSI 通讯的数据包被丢弃

如果防火墙很忙，可能会丢弃包。SUSE 防火墙默认为三分钟后丢弃包。如果您发现 iSCSI 通讯数据包被丢弃，请考虑将 SUSE 防火墙配置为太忙时将数据包排队，而不是丢弃它们。

15.6.4 将 iSCSI 卷与 LVM 配合使用

在 iSCSI 目标上使用 LVM 时，请使用本部分的查错提示。

15.6.4.1 检查在引导时是否执行 iSCSI 发起端发现

设置 iSCSI 发起端时，确保引导时启用发现，以便 udev 在引导时可以发现 iSCSI 设备，并设置可供 LVM 使用的设备。

15.6.4.2 检查在引导时是否执行 iSCSI 目标发现

请记住，udev 会为设备提供默认设置。确保创建设备的所有应用程序均已在引导时启动，以便 **udev** 在系统启动时能为这些应用程序识别出并指派设备。如果应用程序或服务在稍后才启动，则 **udev** 不会像在引导时那样自动创建设备。

15.6.5 配置文件设置为手动时，会挂载 iSCSI 目标

如果您之前手动修改了 `/etc/iscsi/iscsid.conf` 配置文件，那么，即使在该文件中将 `node.startup` 选项设置为手动，Open-iSCSI 启动时也会挂载目标。

检查 `/etc/iscsi/nodes/TARGET_NAME/IP_ADDRESS,PORT/default` 文件。它包含覆盖 `/etc/iscsi/iscsid.conf` 文件的 `node.startup` 设置。如果使用 YaST 接口将挂载选项设置为手动，同时会在 `/etc/iscsi/nodes/TARGET_NAME/IP_ADDRESS,PORT/default` 文件中设置 `node.startup = manual`。

15.7 iSCSI LIO 目标术语

后备储存

提供 iSCSI 端点底层的实际储存的物理储存对象。

CDB（命令描述符块）)

SCSI 命令的标准格式。CDB 通常长度为 6、10 或 12 个字节，但也可以达到 16 个字节或具有可变长度。

CHAP（质询握手身份验证协议）

点对点协议 (PPP) 身份验证方法用于向另一台计算机确认一台计算机的身份。链路控制协议 (LCP) 连接两台计算机并协商 CHAP 方法后，身份验证器会向对等体发送随机询问。然后该对等体会根据询问以及密钥发出以哈希加密的应答。再由身份验证器比照由预期哈希值自己计算出的结果对该哈希应答进行验证，以决定是确认身份验证还是中断连接。CHAP 在 RFC 1994 中定义。

CID（连接标识符）

由发起端生成的 16 位编号，用于唯一识别两个 iSCSI 设备之间的连接。登录阶段会显示此编号。

端点

iSCSI 目标名称与 iSCSI TPG（IQN + 标记）的组合。

EUI（扩展的唯一标识符）

在全球范围内唯一标识每个设备的 64 位编号。其格式由指定公司的唯一的 24 位编号和由该公司分配给所构建的每个设备的 40 位编号组成。

发起端

SCSI 会话的发起端。通常是计算机等控制设备。

IPS（互联网协议储存）

使用 IP 协议在储存网络中移动数据的一类协议或设备。FCIP（基于 IP 的光纤通道）、iFCP（互联网光纤通道协议）以及 iSCSI（互联网 SCSI）都属于 IPS 协议。

IQN（iSCSI 限定的名称）

在全球范围内唯一标识每个设备的 iSCSI 的名称格式（例如：`iqn.5886.com.acme.tapedrive.sn-a12345678`）。

ISID（发起端会话标识符）

由发起端生成的一个 48 位编号，用于唯一识别发起端与目标之间的会话。此值在登录过程中创建，生成后将与登录 PDU 一起发送给目标。

MCS（每个会话多重连接）

它是 iSCSI 规范的组成部分，允许发起端与目标之间存在多个 TCP/IP 连接。

MPIO（多路径 I/O）

一种可在服务器和储存设备之间建立多个冗余数据路径的方法。

网络门户

iSCSI 端点与 IP 地址及 TCP（传输控制协议）端口的组合。TCP 端口 3260 是 IANA（互联网号码分配机构）所定义的 iSCSI 协议的端口号。

SAM（SCSI 体系结构模型）

以一般术语说明 SCSI 行为的文档。它支持不同类型的设备通过各种介质进行通讯。

target

SCSI 会话的接收端，通常是磁盘驱动器、磁带驱动器或扫描仪等设备。

目标组 (TG)

在创建视图时视为同等对待的 SCSI 目标端口的列表。创建视图可帮助简化 LUN（逻辑单元号）映射。每个视图项指定一个目标组、主机组以及一个 LUN。

目标端口

一个 iSCSI 端点与一个或多个 LUN 的组合。

目标端口组 (TPG)

IP 地址和 TCP 端口号的列表，用于确定特定 iSCSI 目标将要侦听的接口。

目标会话标识符 (TSID)

由目标生成的 16 位编号，用于唯一标识发起端和目标之间的会话。此值在登录过程中创建，生成后将与登录响应 PDU（协议数据单元）一起发送给发起端。

15.8 更多信息

iSCSI 协议已面世多年。有许多评论将 iSCSI 与 SAN 解决方案作比较、评测性能，此外还有一份文档说明硬件解决方案。有关详细信息，请参见 http://www.open-iscsi.com/ 上的 Open-iSCSI 项目主页。

此外，请参见 **`iscsiadm`**、**`iscsid`** 的手册页，以及示例配置文件 `/etc/iscsid.conf`。

16 以太网光纤通道储存：FCoE

许多企业数据中心的 LAN 和数据流量都依赖于以太网，而其储存基础设施则依赖于光纤通道网络。开放以太网光纤通道 (FCoE) 发起端软件允许配有以太网适配器的服务器经由以太网网络连接光纤通道储存子系统。而在以前，只允许配有光纤通道适配器的系统经由光纤通道架构进行连接。FCoE 技术通过支持网络融合降低了数据中心的复杂度。这有助于保护您在光纤通道储存基础设施方面的现有投资，并简化网络管理。

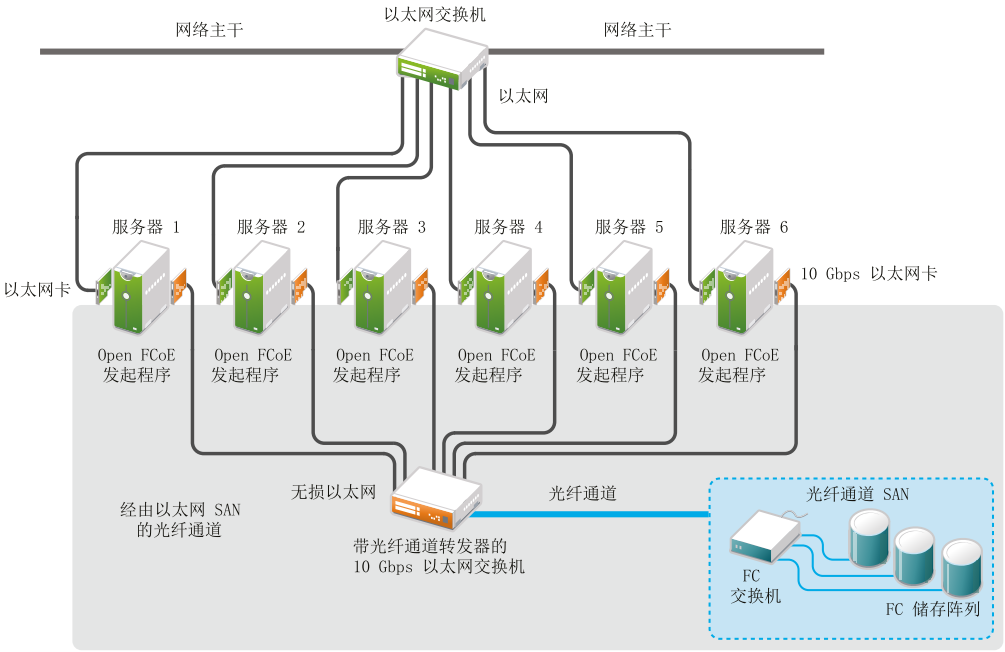


图 16.1： 开放式以太网光纤通道 SAN

Open-FCoE 可让您在主机上而不是主机总线适配器上的专有硬件上运行光纤通道协议。它专用于 10 Gbps（每秒千兆位）以太网适配器，但也可用于任何支持暂停帧的以太网适配器。发起端软件提供光纤通道协议处理模块以及基于以太网的传输模块。Open-FCoE 模块充当 SCSI 的低级驱动程序。Open-FCoE 传输使用 **net_device** 发送和接收包。数据中心桥接 (DCB) 驱动程序提供 FCoE 的服务质量。

FCoE 是一种封装协议，它可以通过以太网连接移动光纤通道协议流量而无需更改光纤通道数据帧。如此一来，您的网络安全和流量管理基础设施就可以像使用光纤通道一样使用 FCoE。

如果存在下列情况，您可以选择在您的企业中部署 FCoE：

- 贵企业已有光纤通道储存子系统，并拥有具备光纤通道技能和知识的管理员。
- 您的网络中部署的是 10 Gbps 以太网。

本章说明如何在您的网络中设置 FCoE。

16.1 在安装过程中配置 FCoE 接口

如果在交换器上为服务器与光纤通道储存基础结构之间的连接启用了 FCoE，则可以通过适用于 SUSE Linux Enterprise Server 的 YaST 来在操作系统安装期间设置 FCoE 磁盘。有些类型的系统 BIOS 能够自动检测到 FCoE 磁盘，并向 YaST 安装软件报告该磁盘。但不是所有类型的 BIOS 都支持 FCoE 磁盘自动检测。在此情况下若要启用自动检测，您可以在开始安装时将 `withfcoe` 选项添加到内核命令行：

```
withfcoe=1
```

当检测到 FCoE 磁盘时，YaST 安装便会在当时提供配置 FCoE 实例的选项。在“磁盘激活”页面上，选择配置 FCoE 接口，以访问 FCoE 配置。有关配置 FCoE 接口的信息，请参见第 16.3 节“使用 YaST 管理 FCoE 服务”。

注意：挂载点支持

引导期间，FCoE 设备将异步显示。虽然 initrd 可确保为根文件系统正确设置这些设备，但对于任何其他文件系统或挂载点（例如 `/usr`），并无此类保证。因此，任何系统挂载点（例如 `/usr` 或 `/var`）都不受支持。要使用这些设备，请确保正确同步相应的服务和设备。

16.2 安装 FCoE 和 YaST FCoE 客户端

您可以通过在交换机上为服务器的连接启用 FCoE，来于储存基础设施中设置 FCoE 磁盘。如果在安装 SUSE Linux Enterprise Server 操作系统时 FCoE 磁盘可用，系统会在那时自动安装 FCoE 发起端软件。

如果 FCoE 发起端软件和 YaST FCoE 客户端软件均未安装，请使用下列程序通过下面的命令手动安装它们：

```
> sudo zypper in yast2-fcoe-client fcoe-utils
```

也可以使用 YaST 软件管理器来安装以上所列软件包。

16.3 使用 YaST 管理 FCoE 服务

您可以使用“YaST FCoE 客户端配置”选项为光纤通道储存基础设施中的 FCoE 磁盘创建、配置和删除 FCoE 接口。若要使用此选项，必须安装并运行 FCoE 发起端服务（`fcoemon` 守护程序）和“链路层发现协议”代理守护程序 (`llpad`)，并且必须在支持 FCoE 的交换器上启用 FCoE 连接。

1. 启动 YaST 并选择网络服务 › FCoE 客户端配置。

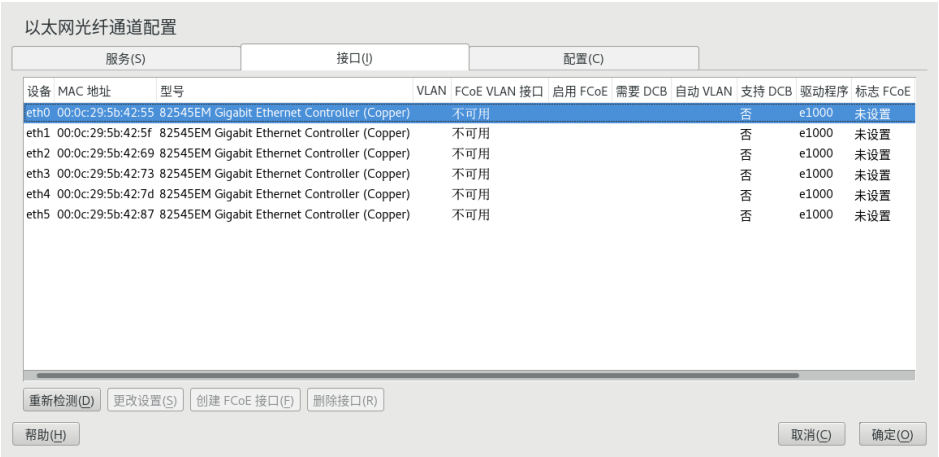
2. 在服务选项卡上，可按需要查看或修改 FCoE 服务和 Lldpad（链路层发现协议代理守护程序）服务启动时间。

   - **启动以太网光纤通道 (FCoE) 服务：** 指定是在服务器引导时启动以太网光纤通道服务 `fcoemon` 守护程序还是手动启动。该守护程序用于控制 FCoE 接口，并可与 `llpad` 守护程序创建连接。可用值为引导时（默认）或手动。

   - **启动 Lldpad 服务：** 指定是在服务器引导时启动“链路层发现协议”代理 `llpad` 守护程序，还是手动启动该守护程序。`llpad` 守护程序会向 `fcoemon` 守护程序报告数据中心桥接功能和 FCoE 接口配置的相关信息。可用值为引导时（默认）或手动。

   如果修改设置，则单击确定保存并应用更改。

3. 在接口选项卡上，可查看有关服务器上所有检测到的网络适配器的信息，包括有关 VLAN 和 FCoE 配置的信息。您也可以创建 FCoE VLAN 接口、更改现有 FCoE 接口的设置，或删除 FCoE 接口。



请使用 FCoE VLAN 接口列来确定 FCoE 是否可用：

接口名

如果将 `eth4.200` 之类名称指派给接口，则 FCoE 在交换机上可用，并会为该适配器激活 FCoE 接口。

未配置：

如果状态为未配置，则表示在交换机上启用 FCoE 但尚未为适配器激活 FCoE 接口。选择适配器，然后单击创建 FCoE VLAN 接口，以在适配器上激活接口。

不可用：

如果状态为不可用，则表示因为没有在交换机上为该连接启用 FCoE，所以 FCoE 对适配器不可用。

4. 若要设置未经设置的支持 FCoE 的适配器，予以选中，然后单击创建 FCoE VLAN 接口。对出现的问题请单击是确认。
   该适配器现在会以某个接口名称列在 FCoE VLAN 接口列中。

5. 若要更改已设置适配器的设置，请在列表中选择它，然后单击更改设置。
   可设置下列选项：

   启用以太网光纤通道

   启用或禁用为适配器创建 FCoE 实例。

   需要数据中心桥接

   指定适配器是否需要数据中心桥接（通常会需要）。

自动 VLAN

指定 `fcoemon` 守护程序是否自动创建 VLAN 接口。

如果修改设置，则单击下一步保存并应用更改。设置将写入 `/etc/fcoe/cfg-ethX` 文件。`fcoemon` 守护程序会在初始化时读取每个 FCoE 接口的配置文件。

6. 若要去除已设置的接口，请从列表中选择它。单击删除接口，然后单击继续确认。FCoE 接口值将更改为未配置。

7. 在配置选项卡上，可查看或修改 FCoE 系统服务的常规设置。您可以从 FCoE 服务脚本和 `fcoemon` 守护程序中启用或禁用调试消息，并可指定是否将消息发送至系统日志。

8. 单击确定保存并应用更改。

16.4 使用命令配置 FCoE

以下步骤需使用 **`fipvlan`** 命令。如果未安装该命令，请运行以下命令来安装：

```
> sudo zypper in fcoe-utils
```

要发现并配置所有以太网接口，请执行以下步骤：

1. 打开终端。
2. 要发现所有可用的以太网接口，请运行以下命令：

   ```
   > sudo fipvlan -a
   ```

3. 对于配置了 FCoE 卸载的每个以太网接口，请运行以下命令：

   ```
   > sudo fipvlan -c -s ETHERNET_INTERFACE
   ```

   如果该网络接口不存在，该命令将创建一个网络接口，并在发现的 FCoE VLAN 上启动 `Open-FCoE` 发起端。

16.5 使用 FCoE 管理工具管理 FCoE 实例

`fcoeadm` 实用程序是以太网光纤通道 (FCoE) 管理工具。可以使用该工具创建、销毁和重置指定网络接口上的 FCoE 实例。**`fcoeadm`** 实用程序通过套接字接口将命令发送给正在运行的 `fcoemon` 进程。有关 **`fcoemon`** 的相关信息，请参见 **`man 8 fcoemon`**。

`fcoeadm` 实用程序可让您查询 FCoE 实例的以下相关信息：

- 接口
- 目标 LUN
- 端口统计数字

`fcoeadm` 实用程序是 `fcoe-utils` 软件包的组成部分，该命令的一般语法如下所示：

```
fcoeadm
  [-c|--create] [<ethX>]
  [-d|--destroy] [<ethX>]
  [-r|--reset] [<ethX>]
  [-S|--Scan] [<ethX>]
  [-i|--interface] [<ethX>]
  [-t|--target] [<ethX>]
```

```
  [-l|--lun] [<ethX>]
  [-s|--stats <ethX>] [<interval>]
  [-v|--version]
  [-h|--help]
```

有关细节，请参见 **man 8 fcoeadm**。

示例

fcoeadm -c eth2.101

在 eth2.101 上创建 FCoE 实例。

fcoeadm -d eth2.101

删除 eth2.101 上的 FCoE 实例。

fcoeadm -i eth3

显示接口 eth3 上所有 FCoE 实例的相关信息。如果未指定任何接口，则会显示所有已创建 FCoE 实例的接口的相关信息。下面的示例显示连接 eth0.201 的相关信息：

```
> sudo fcoeadm -i eth0.201
  Description:      82599EB 10-Gigabit SFI/SFP+ Network Connection
  Revision:         01
  Manufacturer:     Intel Corporation
  Serial Number:    001B219B258C
  Driver:           ixgbe 3.3.8-k2
  Number of Ports:  1

      Symbolic Name:     fcoe v0.1 over eth0.201
      OS Device Name:    host8
      Node Name:         0x1000001B219B258E
      Port Name:         0x2000001B219B258E
      FabricName:        0x2001000573D38141
      Speed:             10 Gbit
      Supported Speed:   10 Gbit
      MaxFrameSize:      2112
      FC-ID (Port ID):   0x790003
      State:             Online
```

fcoeadm -l eth3.101

显示在连接 eth3.101 上发现到的所有 LUN 的详细信息。如果未指定任何连接，则会显示在所有 FCoE 连接上发现到的所有 LUN 的相关信息。

fcoeadm -r eth2.101

重设置 eth2.101 上的 FCoE 实例。

fcoeadm -s eth3 3

以三秒的时间间隔显示具有 FCoE 实例的特定 eth 3 端口的统计信息。每隔一段时间会显示一行统计数字。若未指定时间间隔，则使用默认值一秒。

fcoeadm -t eth3

显示从具有 FCoE 实例的指定 eth3 端口发现到的所有目标的相关信息。每一个已发现目标后面列出了所有关联的 LUN。若未指定实例，则显示来自具有 FCoE 实例的所有端口的目标。下面的示例显示来自 eth0.201 连接的目标的相关信息：

```
> sudo fcoeadm -t eth0.201
  Interface:        eth0.201
  Roles:            FCP Target
  Node Name:        0x200000D0231B5C72
  Port Name:        0x210000D0231B5C72
  Target ID:        0
  MaxFrameSize:     2048
  OS Device Name:   rport-8:0-7
  FC-ID (Port ID):  0x79000C
  State:            Online

LUN ID  Device Name   Capacity   Block Size  Description
------  -----------  ----------  ----------  ----------------------------
    40  /dev/sdqi     792.84 GB      512     IFT DS S24F-R2840-4 (rev 386C)
    72  /dev/sdpk     650.00 GB      512     IFT DS S24F-R2840-4 (rev 386C)
   168  /dev/sdgy       1.30 TB      512     IFT DS S24F-R2840-4 (rev 386C)
```

16.6 更多信息

有关信息，请参见以下文档：

- 有关 Open-FCoE 服务守护程序的信息，请参见 **`fcoemon(8)`** 手册页。
- 有关 Open-FCoE 管理工具的信息，请参见 **`fcoeadm(8)`** 手册页。
- 有关数据中心桥接配置工具的信息，请参见 **`dcbtool(8)`** 手册页。
- 有关链路层发现协议代理守护程序的信息，请参见 `lldpad(8)` 手册页。

17 NVMe over Fabric

本章介绍如何设置 NVMe over Fabric 主机和目标。

17.1 概览

NVM Express (**NVMe**) 是有关访问非易失性储存（通常是 SSD 磁盘）的接口标准。与 SATA 相比，NVMe 支持的速度要高得多，并且延迟更低。

NVMe over Fabric 是用于通过不同网络结构访问 NVMe 储存的体系结构。这些网络结构有 **RDMA**、**TCP** 或**基于光纤通道的 NVMe** (**FC-NVMe**) 等。NVMe over Fabric 的作用类似于 iSCSI。为提高容错能力，NVMe over Fabric 内置了多路径支持。NVMe over Fabric 多路径不是基于传统的 DM 多路径。

NVMe 主机是指连接到 NVMe 目标的计算机。**NVMe 目标**是指共享其 NVMe 块设备的计算机。

SUSE Linux Enterprise Server 15 SP4 支持 NVMe。提供了可用于 NVMe 块储存以及 NVMe over Fabric 目标和主机的内核模块。

要查看您的硬件是否有任何特殊注意事项，请参见第 17.4 节 “特殊硬件配置”。

17.2 设置 NVMe over Fabric 主机

要使用 NVMe over Fabric，必须在目标上使用支持的联网方法。支持的方法包括基于光纤通道的 NVMe、TCP 和 RDMA。以下几节介绍如何将主机连接到 NVMe 目标。

17.2.1 安装命令行客户端

要使用 NVMe over Fabric，需要安装 **nvme** 命令行工具。请使用 **zypper** 安装该工具：

```
> sudo zypper in nvme-cli
```

使用 **nvme --help** 可列出所有可用的子命令。我们提供了 **nvme** 子命令的手册页。执行 **man nvme-**SUBCOMMAND 可查阅相关的手册页。例如，要查看 discover 子命令的手册页，请执行 **man nvme-discover**。

17.2.2 发现 NVMe over Fabric 目标

要列出 NVMe over Fabric 目标上的可用 NVMe 子系统，您需要有发现控制器地址和服务 ID。

```
> sudo nvme discover -t TRANSPORT -a DISCOVERY_CONTROLLER_ADDRESS -s SERVICE_ID
```

将 TRANSPORT 替换为底层传输媒体：loop、rdma、tcp 或 fc。将 DISCOVERY_CONTROLLER_ADDRESS 替换为发现控制器的地址。对于 RDMA 和 TCP，此地址应是 IPv4 地址。将 SERVICE_ID 替换为传输服务 ID。如果服务基于 IP（例如 RDMA 或 TCP），则服务 ID 指定端口号。对于光纤通道，不需要提供服务 ID。

NVMe 主机只会看到它们有权连接的子系统。

示例：

```
> sudo nvme discover -t tcp -a 10.0.0.3 -s 4420
```

有关更多细节，请参见 **man nvme-discover**。

17.2.3 连接 NVMe over Fabric 目标

识别 NVMe 子系统后，便可以使用 **nvme connect** 命令来连接它。

```
> sudo nvme connect -t transport -a DISCOVERY_CONTROLLER_ADDRESS -s SERVICE_ID -
n SUBSYSTEM_NQN
```

将 TRANSPORT 替换为底层传输媒体：loop、rdma、tcp 或 fc。将 DISCOVERY_CONTROLLER_ADDRESS 替换为发现控制器的地址。对于 RDMA 和 TCP，此地址应是 IPv4 地址。将 SERVICE_ID 替换为传输服务 ID。如果服务基于 IP（例如 RDMA 或 TCP），则此 ID 指定端口号。将 SUBSYSTEM_NQN 替换为发现命令找到的所需子系统的 NVMe 限定名称。**NQN** 是 **NVMe 限定名称** (NVMe Qualified Name) 的缩写。NQN 必须唯一。

示例：

```
> sudo nvme connect -t tcp -a 10.0.0.3 -s 4420 -n
 nqn.2014-08.com.example:nvme:nvm-subsystem-sn-d78432
```

或者，使用 **nvme connect-all** 连接到发现的所有名称空间。有关高级用法，请参见 **man nvme-connect** 和 **man nvme-connect-all**。

如果发生路径丢失，NVMe 子系统会尝试重新连接一段时间，该时间由 **`nvme connect`** 命令的 `ctrl-loss-tmo` 选项定义。在这段时间（默认值为 600s）过后，将去除该路径并通知块层（文件系统）的上层。默认情况下，随后会以只读模式挂载文件系统，这通常不是预期的行为。因此，建议设置 `ctrl-loss-tmo` 选项，使 NVMe 子系统无限制地持续尝试重新连接。为此，请运行以下命令：

```
> sudo nvme connect --ctrl-loss-tmo=-1
```

要使 NVMe over Fabric 子系统在引导时可用，请在主机上创建一个 `/etc/nvme/discovery.conf` 文件，并在其中包含传递给 **`discover`** 命令的参数（参见第 17.2.2 节 “发现 NVMe over Fabric 目标”）。例如，如果您按如下所示使用 **`discover`** 命令：

```
> sudo nvme discover -t tcp -a 10.0.0.3 -s 4420
```

请将 **`discover`** 命令的参数添加到 `/etc/nvme/discovery.conf` 文件中：

```
echo "-t tcp -a 10.0.0.3 -s 4420" | sudo tee -a /etc/nvme/discovery.conf
```

然后启用 nvmf-autoconnect 服务：

```
> sudo systemctl enable nvmf-autoconnect.service
```

17.2.4 多路径

NVMe 本机多路径默认处于启用状态。如果在控制器标识设置中设置了 `CMIC` 选项，则 NVMe 堆栈默认会将 NVMe 驱动器识别为多路径设备。

要管理多路径，您可以使用以下命令：

管理多路径

`nvme list-subsys`
: 打印多路径设备的布局。

`multipath -ll`
: 命令 有兼容模式，可显示 NVMe 多路径设备。

`nvme-core.multipath=N`
: 当作为引导参数添加该选项时，将禁用 NVMe 本机多路径。

注意：在多路径设置中使用 iostat

iostat 命令可能不会显示 **nvme list-subsys** 列出的所有控制器。默认情况下，**iostat** 会过滤掉所有没有 I/O 的块设备。要使 **iostat** 显示**所有**设备，请使用：

```
iostat -p ALL
```

17.3 设置 NVMe over Fabric 目标

17.3.1 安装命令行客户端

要配置 NVMe over Fabric 目标，需要安装 **nvmetcli** 命令行工具。请使用 **zypper** 安装该工具：

```
> sudo zypper in nvmetcli
```

http://git.infradead.org/users/hch/nvmetcli.git/blob_plain/HEAD:/Documentation/nvmetcli.txt 上提供了 **nvmetcli** 的最新文档。

17.3.2 配置步骤

下面的过程举例说明如何设置 NVMe over Fabric 目标。

配置储存在树型结构中。使用 **cd** 命令可导航。使用 **ls** 可列出对象。您可以使用 **create** 创建新对象。

1. 启动 **nvmetcli** 交互外壳：

   ```
   > sudo nvmetcli
   ```

2. 创建新端口：

   ```
   (nvmetcli)> cd ports
   (nvmetcli)> create 1
   ```

```
(nvmetcli)> ls 1/
o- 1
  o- referrals
  o- subsystems
```

3. 创建 NVMe 子系统：

```
(nvmetcli)> cd /subsystems
(nvmetcli)> create
 nqn.2014-08.org.nvmexpress:NVMf:uuid:c36f2c23-354d-416c-95de-f2b8ec353a82
(nvmetcli)> cd
 nqn.2014-08.org.nvmexpress:NVMf:uuid:c36f2c23-354d-416c-95de-f2b8ec353a82/
(nvmetcli)> ls
o- nqn.2014-08.org.nvmexpress:NVMf:uuid:c36f2c23-354d-416c-95de-
f2b8ec353a82
  o- allowed_hosts
  o- namespaces
```

4. 创建新的名称空间，并在其中设置 NVMe 设备：

```
(nvmetcli)> cd namespaces
(nvmetcli)> create 1
(nvmetcli)> cd 1
(nvmetcli)> set device path=/dev/nvme0n1
Parameter path is now '/dev/nvme0n1'.
```

5. 启用先前创建的名称空间：

```
(nvmetcli)> cd ..
(nvmetcli)> enable
The Namespace has been enabled.
```

6. 显示创建的名称空间：

```
(nvmetcli)> cd ..
(nvmetcli)> ls
o- nqn.2014-08.org.nvmexpress:NVMf:uuid:c36f2c23-354d-416c-95de-
f2b8ec353a82
  o- allowed_hosts
  o- namespaces
```

```
    o- 1
```

7. 允许所有主机使用该子系统。只有在安全环境中才可这样做。

```
(nvmetcli)> set attr allow_any_host=1
Parameter allow_any_host is now '1'.
```

或者，可以只允许特定的主机建立连接：

```
(nvmetcli)> cd
 nqn.2014-08.org.nvmexpress:NVMf:uuid:c36f2c23-354d-416c-95de-f2b8ec353a82/
allowed_hosts/
(nvmetcli)> create hostnqn
```

8. 列出创建的所有对象：

```
(nvmetcli)> cd /
(nvmetcli)> ls
o- /
  o- hosts
  o- ports
  | o- 1
  |   o- referrals
  |   o- subsystems
  o- subsystems
    o- nqn.2014-08.org.nvmexpress:NVMf:uuid:c36f2c23-354d-416c-95de-
f2b8ec353a82
      o- allowed_hosts
      o- namespaces
        o- 1
```

9. 使目标可通过 TCP 使用。为 RDMA 使用 `trtype=rdma`：

```
(nvmetcli)> cd ports/1/
(nvmetcli)> set addr adrfam=ipv4 trtype=tcp traddr=10.0.0.3 trsvcid=4420
Parameter trtype is now 'tcp'.
Parameter adrfam is now 'ipv4'.
Parameter trsvcid is now '4420'.
Parameter traddr is now '10.0.0.3'.
```

或者，使目标可通过光纤通道使用：

```
(nvmetcli)> cd ports/1/
(nvmetcli)> set addr adrfam=fc trtype=fc
 traddr=nn-0x1000000044001123:pn-0x2000000055001123 trsvcid=none
```

10. 将子系统链接到该端口：

```
(nvmetcli)> cd /ports/1/subsystems
(nvmetcli)> create
 nqn.2014-08.org.nvmexpress:NVMf:uuid:c36f2c23-354d-416c-95de-f2b8ec353a82
```

现在，可以使用 **dmesg** 校验该端口是否已启用：

```
# dmesg
        ...
[  257.872084] nvmet_tcp: enabling port 1 (10.0.0.3:4420)
```

17.3.3 备份和恢复目标配置

可使用以下命令在 JSON 文件中保存目标配置：

```
> sudo nvmetcli
(nvmetcli)> saveconfig nvme-target-backup.json
```

要恢复配置，请使用：

```
(nvmetcli)> restore nvme-target-backup.json
```

您还可以擦除当前配置：

```
(nvmetcli)> clear
```

17.4 特殊硬件配置

17.4.1 概览

有些硬件需要特殊的配置才能正常工作。请浏览以下章节的标题，确定您是否使用了所提到的设备或供应商。

17.4.2 Broadcom

如果您在使用 **Broadcom Emulex LightPulse Fibre Channel SCSI** 驱动程序，请在 `lpfc` 模块的目标和主机中添加内核配置参数：

```
> sudo echo "options lpfc lpfc_enable_fc4_type=3" > /etc/modprobe.d/lpfc.conf
```

确保 Broadcom 适配器固件的版本至少为 11.4.204.33。另外确保已安装最新版本的 `nvme-cli`、`nvmetcli` 和内核。

要启用光纤通道端口作为 NVMe 目标，需要额外配置一个模块参数：`lpfc_enable_nvmet=COMMA_SEPARATED_WWPNS`。请输入带有前置 `0x` 的 WWPN，例如 `lpfc_enable_nvmet=0x2000000055001122,0x2000000055003344`。系统只会为目标模式配置列出的 WWPN。可将光纤通道端口配置为目标**或**发起端。

17.4.3 Marvell

QLE269x 和 QLE27xx 适配器上都支持 FC-NVMe。Marvell® QLogic® QLA2xxx 光纤通道驱动程序中默认会启用 FC-NVMe 支持。

要确认 NVMe 是否已启用，请运行以下命令：

```
> cat /sys/module/qla2xxx/parameters/ql2xnvmeenable
```

如果显示 `1`，则表示 NVMe 已启用，显示 `0` 表示 NVMe 已禁用。

然后，检查以下命令的输出，确保 Marvell 适配器固件的版本不低于 8.08.204：

```
> cat /sys/class/scsi_host/host0/fw_version
```

最后，确保已安装适用于 SUSE Linux Enterprise Server 的最新版 `nvme-cli`、`QConvergeConsoleCLI` 和内核。例如，您可以运行

```
# zypper lu && zypper pchk
```

来检查更新和补丁。

有关安装的更多细节，请参考下列 Marvell 用户指南中的 FC-NVMe 部分：

- http://driverdownloads.qlogic.com/QLogicDriverDownloads_UI/ShowEula.aspx?resourceid=32769&docid=96728&ProductCategory=39&Product=1259&Os=126
- http://driverdownloads.qlogic.com/QLogicDriverDownloads_UI/ShowEula.aspx?resourceid=32761&docid=96726&ProductCategory=39&Product=1261&Os=126

17.5 更多信息

有关 **`nvme`** 命令的功能的更多细节，请参考 **`nvme nvme-help`**。

以下链接提供了 NVMe 和 NVMe over Fabric 的简介：

- http://nvmexpress.org/
- http://www.nvmexpress.org/wp-content/uploads/NVMe_Over_Fabrics.pdf
- https://storpool.com/blog/demystifying-what-is-nvmeof

18 管理设备的多路径 I/O

本章介绍如何使用多路径 I/O (MPIO) 来管理服务器和块储存设备间多路径的故障转移和路径负载平衡。

18.1 了解多路径 I/O

多路径是服务器与跨多个物理路径（这些路径在服务器中的主机总线适配器和设备储存控制器之间）的同一物理或逻辑块储存设备通讯的能力，通常是在光纤通道 (FC) 或 iSCSI SAN 环境中。

Linux 的多路径提供连接容错，并可以跨多个活动连接提供负载平衡。当多路径已配置并且正在运行时，它会自动隔离和识别设备连接故障，并重路由 I/O 以改变连接。

多路径针对连接故障提供容错能力，但不针对储存设备本身的故障提供容错能力。针对后者的容错是通过镜像等互补技术实现的。

18.1.1 多路径术语

储存阵列

包含许多磁盘和多个结构连接（控制器）的硬件设备，为客户端提供 SAN 或 NAS 储存。储存阵列通常具备 RAID 和故障转移功能并支持多路径。一直以来，主动/被动（故障转移）和主动/主动（负载平衡）储存阵列的配置是有区别的。这些概念仍然存在，但它们不过是新式硬件所支持的路径组和访问状态概念的特殊情况。

主机、主机系统

运行 SUSE Linux Enterprise Server 的计算机，充当**储存阵列**的客户端系统。

多路径映射、多路径设备

一组**路径设备**。它代表储存阵列上的储存卷，被主机系统视为单个块设备。

路径设备、低级设备

多路径映射的成员，通常是一个 SCSI 设备。每个路径设备代表主机计算机与实际储存卷之间的唯一连接，例如，来自 iSCSI 会话的逻辑单元。在 Linux 设备映射程序多路径下，路径设备在主机系统中保持可见且可访问。

WWID、UID、UUID

“全球标识符”、“唯一标识符”、“全局唯一标识符”。WWID 是储存卷的一个属性，因此，它在多路径映射的所有路径设备之间是相同的。`multipath-tools` 使用 WWID 来确定应将哪些低级设备汇编到多路径映射中。多路径依赖使用 **udev** 来确定路径设备的 WWID。多路径映射的 WWID 永远不会更改。可以通过 `/dev/disk/by-id/dm-uuid-mpath-WWID` 可靠地访问多路径设备。

WWID 应该与可配置的**映射名称**相区分（请参见第 18.9 节 “配置用户用好的名称或别名”）。

uevent、udev 事件

由内核发送到用户空间并由 **udev** 子系统处理的事件。在添加、去除设备或更改其属性时会生成 uevent。

设备映射程序

Linux 内核中用于创建虚拟块设备的框架。被映射设备的 I/O 操作将重定向到底层块设备。设备映射可以堆叠。设备映射程序实现自身的事件信令（也称为“设备映射程序事件”或“dm 事件”）。

18.2 硬件支持

多路径驱动程序和工具可在 SUSE Linux Enterprise Server 支持的所有体系结构上使用。协议无关的通用驱动程序适用于市场上大多数支持多路径的储存硬件。某些储存阵列供应商提供自己的多路径管理工具。请查看供应商的硬件文档以确定需要哪些设置。

18.2.1 多路径实现：设备映射程序和 NVMe

Linux 中的传统通用多路径实现使用设备映射程序框架。对于 SCSI 设备等大多数设备类型，设备映射程序多路径是唯一可用的实现。设备映射程序多路径具有很高的可配置性和灵活性。

Linux **NVM Express** (NVMe) 内核子系统在内核中本机实现多路径。这种实现可以降低 NVMe 设备（通常是延迟极低的快速设备）的计算开销。本机 NVMe 多路径不需要用户空间组件。从 SLE 15 开始，本机多路径一直是 NVMe 多路径设备的默认设置。

本章的余下内容介绍设备映射程序多路径。

18.2.2 针对多路径的储存阵列自动检测

设备映射程序多路径是一种通用技术。多路径设备检测只要求内核检测低级（例如 SCSI）设备，并要求设备属性可靠地将多个低级设备标识为同一个卷的不同“路径”，而不是实际不同的设备。

`multipath-tools` 软件包按供应商和产品名称检测储存阵列。它已验证多种不同储存产品的内置配置默认值。请查阅储存阵列的硬件文档；某些供应商为 Linux 多路径配置提供了具体的建议。要查看在系统上检测到的内置储存设置，请运行命令 **`multipath -T`**，具体请参见第 18.4.5 节 “multipath 命令”。

如果您需要对储存阵列的内置配置应用更改，请创建并配置 `/etc/multipath.conf` 文件。请参见第 18.6 节 “多路径配置”。

注意

`multipath-tools` 为许多储存阵列提供内置预设。给定储存产品存在此类预设**并不意味着**该储存产品的供应商已使用 `dm-multipath` 测试了该产品，也不意味着该供应商认可或支持对该产品使用 `dm-multipath`。如有支持相关的问题，请始终查阅原始供应商文档。

18.2.3 需要特定硬件处理程序的储存阵列

某些储存阵列要求运行特殊命令才能从一条路径故障转移到另一条路径，或要求使用非标准的错误处理方法。这些特殊命令和方法由 Linux 内核中的硬件处理程序实现。新式 SCSI 储存阵列支持 SCSI 标准中定义的“非对称逻辑单元访问”(ALUA) 硬件处理程序。除 ALUA 之外，SLE 内核还包含 Netapp E 系列 (RDAC)、Dell/EMC CLARiiON CX 阵列系列和 HP 传统阵列的硬件处理程序。从 Linux 内核 4.4 开始，Linux 内核已自动检测到大多数阵列（包括所有支持 ALUA 的阵列）的硬件处理程序。

18.3 规划多路径

当规划多路径 I/O 解决方案时，请使用本节中的准则。

18.3.1 先决条件

- 用于多路径设备的储存阵列必须支持多路径。有关详细信息，请参见第 18.2 节 “硬件支持”。
- 仅当在服务器中的主机总线适配器和块储存设备的主机总线控制器之间存在多个物理路径时，才需要配置多路径。
- 对于某些储存阵列，供应商提供其自己的多路径软件以管理该阵列物理和逻辑设备的多路径。在这种情况下，您应遵循供应商关于为这些设备配置多路径的说明。
- 当在虚拟化环境中使用多路径时，多路径在主机服务器环境中控制。先配置设备的多路径，再将其指派给虚拟 Guest 计算机。

18.3.2 多路径安装类型和 initramfs

18.3.2.1 根文件系统位于多路径上 (SAN-boot)

根文件系统位于多路径设备上（通常，所有其他文件系统也位于多路径储存上）。对于仅使用 SAN 或 NAS 储存的无磁盘服务器，情况往往如此。在此类系统上，需要支持多路径才能完成引导，并且必须在 initramfs (initrd) 中启用多路径。请参见第 18.3.2.3 节 “使初始 RAM 磁盘保持同步”。

18.3.2.2 根文件系统位于本地磁盘上

根文件系统（可能还包括其他某些文件系统）位于本地储存中，例如，位于直接挂接的 SATA 磁盘或本地 RAID 上，但系统另外使用多路径 SAN 或 NAS 储存中的文件系统。可以通过三种不同的方式配置这种系统类型：

使用 root-on-multipath 设置

所有块设备（包括本地磁盘）是多路径映射的一部分。然后，此类设置将显示为只有一条路径的降级多路径映射。如果在使用 YaST 进行初始系统安装期间启用了多路径，则会创建此配置。这是最简单的配置，但会产生性能开销。

`multipath-tools` **忽略本地磁盘**

在此配置中，多路径是在 initramfs 中启用的。可以在安装后通过黑名单或 `find_multipaths` 配置参数实现此配置。

在 initramfs 中禁用多路径

如果在使用 YaST 进行初始系统安装期间未启用多路径（原因是 YaST 未检测到多路径设备，或者用户在安装期间选择不启用多路径），则会创建此设置。这是 第 18.3.2.3 节 “使初始 RAM 磁盘保持同步” 不适用的唯一一种情况。

18.3.2.3 使初始 RAM 磁盘保持同步

重要

确保初始 RAM 磁盘和引导的系统在对所有块设备使用多路径方面保持一致行为。应用多路径配置更改后重构建 initramfs。

如果在系统中启用了多路径，那么也需要在 `initramfs` 中启用多路径，反之亦然。此规则的唯一例外情况是第 18.3.2.2 节 “根文件系统位于本地磁盘上”中所述的选项 在 initramfs 中禁用多路径。

多路径配置必须在引导的系统与 initrd 之间同步。因此，如果 `/etc/multipath.conf`、`/etc/multipath/wwids`、`/etc/multipath/bindings` 或其他与设备标识相关的配置文件或 udev 规则发生更改，则需要使用以下命令重构建初始 RAM FS：

```
> sudo dracut -f
```

如果 `initrd` 与系统不同步，系统将无法正常引导，启动过程可能显示紧急外壳。有关如何避免或修复此类情况的说明，请参见第 18.15.2 节 “启用多路径时系统在引导过程中退出到紧急外壳”。

如果要在非一般情况下重构建初始 RAM 磁盘（例如，从救援系统重构建，或使用内核参数 `multipath=off` 引导后重构建），必须格外小心。当且仅当 **`dracut`** 在构建 initrd 期间检测到根文件系统位于多路径设备上时，它才会自动在初始 RAM 磁盘中包含多路径支持。在这种情况下，需要显式启用或禁用多路径。

要在 `initrd` 中启用多路径支持，请运行以下命令：

```
> sudo dracut --force --add multipath
```

要在 `initrd` 中禁用多路径支持，请运行以下命令：

```
> sudo dracut --force --omit multipath
```

18.3.3 磁盘管理任务

使用第三方 SAN 阵列管理工具或储存阵列的用户界面来创建逻辑设备，并将其指派到主机。确保在两端正确配置主机身份凭证。

可以在正在运行的主机中添加或去除卷，但检测更改可能需要重新扫描 SCSI 目标并在主机上重新配置多路径。

18.3.4 软件 RAID 和复杂的储存堆栈

多路径是在 SCSI 磁盘等基本储存设备的顶层设置的。在多层储存堆栈中，多路径始终位于底层。其他层（例如软件 RAID、逻辑卷管理、块设备加密等）排布在多路径之上。因此，对于具有多个 I/O 路径以及您计划在软件 RAID 中使用的每个设备，必须将该设备配置为支持多路径，才能尝试创建软件 RAID 设备。

有关为现有软件 RIAD 设置多路径的信息，请参见第 18.12 节 “为现有软件 RAID 配置多路径 I/O”。

18.3.5 高可用性解决方案

群集储存资源的高可用性解决方案基于每个节点上的多路径服务运行。确保每个节点上的 `/etc/multipath.conf` 文件中的配置设置在整个群集中保持一致。

确保多路径设备在所有设备中的名称都相同。有关细节，请参考第 18.9.1 节 “HA 群集中的多路径设备名称”。

用于跨 LAN 镜像设备的分布式复制块设备 (DRBD) 高可用性解决方案在多路径的基础上运行。对于具有多个 I/O 路径并且您计划在 DRDB 解决方案中使用的每个设备，必须先将该设备配置为支持多路径，再配置 DRBD。

将多路径与依赖于使用共享储存实现屏蔽的群集软件（例如包含 **sbd** 的 **pacemaker**）一起使用时必须格外小心。有关详细信息，请参见 第 18.7 节 “配置巡回检测、排队及故障回复的策略”。

18.4 多路径管理工具

SUSE Linux Enterprise Server 中的多路径支持以 Linux 内核的设备映射程序多路径模块及 `multipath-tools` 用户空间软件包为基础。您可以使用多设备管理实用程序 (**multipath**) 查看多路径设备的状态。

18.4.1 设备映射程序多路径模块

设备映射程序多路径 (DM-MP) 模块为 Linux 提供通用多路径功能。DM-MPIO 是 SUSE Linux Enterprise Server 中适用于 SCSI 和 DASD 设备的首选多路径解决方案，它也适用于 NVMe 设备。

注意：将 DM-MP 用于 NVMe 设备

从 SUSE Linux Enterprise Server 15 开始，建议将本机 NVMe 多路径（请参见第 18.2.1 节 “多路径实现：设备映射程序和 NVMe”）用于 NVMe，并且默认情况下会使用它。要禁用本机 NVMe 多路径并改用设备映射程序多路径，请使用内核参数 `nvme-core.multipath=0` 引导。

DM-MPIO 可以为各种设置自动配置多路径子系统。

多路径守护程序 **multipathd** 负责自动路径发现和分组以及自动路径再测试，以便先前发生故障的路径在恢复正常后可以自动重新启用。这可以最大程度地降低管理员对生产环境的关注。

DM-MPIO 保护设备路径以及设备本身不会出故障。如果一个活动路径丢失（例如，一个网络适配器 I/O 断开或光纤电缆被删除），I/O 会重定向到剩余的路径。如果所有活动路径均发生故障，则必须唤醒不活动的辅助路径，因此故障转移最多会延迟 30 秒，具体取决于储存阵列的属性。

如果指向给定设备的每条路径都发生故障，则发往此设备的 I/O 可能会在内核中排队给定的时间，甚至无限期排队（在这种情况下，排队的 IO 总量会受到系统内存的限制）。

如果一个磁盘阵列有多个储存处理器，请确保 SAN 交换机连接到拥有要访问的 LUN 的储存处理器。在多数磁盘阵列上，所有 LUN 均属于这两个储存处理器，所以两个连接都是活动的。

注意：储存处理器

在某些磁盘阵列上，储存阵列通过储存处理器管理通讯，以便一次仅显示一个储存处理器。一个处理器是活动的，另一个是不活动的，直到发生故障。如果您连接到错误的储存处理器（具有不活动路径的那个），则可能无法看到所需的 LUN，或者可能看到这些 LUN，但在尝试访问它们时会出错。

18.4.2 多路径 I/O 管理工具

`multipath-tools` 和 `kpartx` 软件包提供了负责执行自动路径发现和分组的工具。

`multipathd`

用于设置和监视多路径映射的守护程序，以及用来与守护程序进程通讯的命令行客户端。请参见第 18.4.4 节 “**`multipathd`** 守护程序和 **`multipath`** 命令”。

`multipath`

用于执行多路径操作的命令行工具。请参见第 18.4.5 节 “multipath 命令”。

`kpartx`

用于管理多路径设备上的“分区”的命令行工具。请参见第 18.5.3 节 “多路径设备上的分区”。

`mpathpersist`

用于管理 SCSI 永久保留的命令行工具。请参见第 18.4.6 节 “mpathpersist 实用程序”。

18.4.3 多路径设备上的 MD RAID

多路径上的 MD RAID 阵列是由系统的 udev 规则自动设置的。无需在 `/etc/mdadm.conf` 中进行特殊配置。

18.4.4 multipathd 守护程序和 multipath 命令

multipathd 是新式 Linux 设备映射程序多路径设置中的最重要部分。此守护程序通常是通过 systemd 服务 `multipathd.service` 启动的。支持通过 `multipathd.socket` 激活套接字，但**强烈建议**在包含多路径硬件的系统上启用 `multipathd.service`。

multipathd 可处理以下任务（其中一些任务依赖于配置）：

- 启动时，检测路径设备并设置从检测到的设备的多路径映射。
- 监视 uevent 和设备映射程序事件，根据需要在多路径映射中添加或去除路径映射，并启动故障转移或故障回复操作。
- 发现新的路径设备时立即设置新映射。
- 定期检查路径设备以检测故障，并测试有故障的路径，以便在它们恢复正常时重新启用它们。
- 如果所有路径都发生故障，**multipathd** 将使映射失败，或者将映射设备切换到排队模式并让其排队给定的一段时间。
- 处理路径状态更改，并根据需要切换路径组或重新组合路径。
- 测试路径的“边际”状态，即导致路径状态在正常运行和非正常运行之间来回变化的不稳定结构状况。
- 处理路径设备的 SCSI 永久保留密钥（如果已配置）。请参见第 18.4.6 节 “mpathpersist 实用程序”。

multipathd 还充当命令行客户端，它通过将交互式命令发送到正在运行的守护程序来处理这些命令。用于向守护程序发送命令的一般语法如下：

```
multipathd COMMAND
```

或

```
multipathd -k"COMMAND"
```

要进入与守护程序交互的模式，请运行：

```
multipathd -k
```

注意：multipath 和 multipathd 如何配合工作

许多 **multipathd** 命令都有等效的 **multipath** 命令。例如，**multipathd show topology** 的作用与 **multipath -ll** 相同。两者的显著差别在于，multipathd 命令会查询正在运行的 **multipathd** 守护程序的内部状态，而 multipath 则是直接从内核和 I/O 操作获取信息。

如果多路径守护程序正在运行，建议使用 **multipathd** 命令对系统进行修改。否则，守护程序可能会注意到配置更改并对其做出反应。在某些情况下，守护程序甚至可能尝试撤消已应用的更改。因此，如果检测到正在运行的守护程序，**multipath** 会自动将某些可能带来风险的命令（例如销毁和刷新映射）委派给 **multipathd**。

下面的列表介绍了常用的 **multipathd** 命令：

show topology
: 显示当前映射拓扑和属性。

show paths
: 显示当前已知的路径设备。

show paths format "`FORMAT STRING`**"**
: 使用格式字符串显示当前已知的路径设备。使用 **show wildcards** 可查看支持的格式说明符列表。

show maps
: 显示当前配置的映射设备。

show maps format `FORMAT STRING`
: 使用格式字符串显示当前配置的映射设备。使用 **show wildcards** 可查看支持的格式说明符列表。

show config local
: 显示 multipathd 当前正在使用的配置。

reconfigure

重新读取配置文件、重新扫描设备，并再次设置映射。这基本上等同于重启动 **multipathd**。有几个选项在不重启动的情况下无法修改。手册页 multipath.conf(5) 中提到了这些选项。reconfigure 命令只会重新装载以某种形式发生更改的映射设备。要强制重新装载每个映射设备，请使用 **reconfigure all**。

del map MAP DEVICE NAME

取消配置并删除给定的映射设备及其分区。MAP DEVICE NAME 可以是设备节点名称（例如 dm-0）、WWID 或映射名称。如果该设备正在使用中，则该命令将会失败。

可以使用其他命令来修改路径状态、启用或禁用队列等。有关详细信息，请参见 multipathd(8)。

18.4.5 multipath 命令

尽管多路径设置大部分是自动完成的并由 **multipathd** 处理，但 **multipath** 对于某些管理任务仍然很有用。下面提供了几个命令用法示例：

multipath

检测路径设备并配置找到的所有多路径映射。

multipath -d

类似于 **multipath**，但不设置任何映射（“试运行”）。

multipath DEVICENAME

配置特定多路径设备。DEVICENAME 可以按设备节点名称 (/dev/sdb) 或者按主要:次要编号格式的设备编号来表示成员路径设备。或者，它可以是多路径映射的 WWID 或名称。

multipath -f DEVICENAME

取消配置（“刷新”）多路径映射及其分区映射。如果该映射或其某个分区正在使用中，该命令将会失败。有关 DEVICENAME 的可能值，请参见上文。

multipath -F

取消配置（“刷新”）所有多路径映射及其分区映射。如果这些映射正在使用中，该命令将会失败。

multipath -ll

显示所有当前配置的多路径设备的状态和拓扑。

multipath -ll DEVICENAME

显示特定多路径设备的状态。有关 DEVICENAME 的可能值，请参见上文。

multipath -t

显示内部硬件表和活动的多路径配置。有关配置参数的细节，请参见 multipath.conf(5)。

multipath -T

功能与 **multipath -t** 命令类似，但仅显示在主机上检测到的硬件的硬件项。

选项 -v 控制输出的详细程度。您可以使用介于 0（仅限致命错误）和 4（详细日志记录）之间的值。默认值为 -v2。可以在 /etc/multipath.conf 中使用 verbosity 选项来更改 **multipath** 和 **multipathd** 的默认详细程度。

18.4.6 mpathpersist 实用程序

mpathpersist 实用程序用于管理设备映射程序多路径设备上的 SCSI 永久保留。永久保留用于仅限特定的 SCSI 发起端访问 SCSI 逻辑单元。在多路径配置中，必须对给定卷的所有 I_T 节点（路径）使用相同的保留密钥；否则，在一条路径上创建保留可能会导致其他路径发生故障。

将此实用程序与 /etc/multipath.conf 文件中的 reservation_key 属性配合使用可以设置 SCSI 设备的永久保留。当且仅当设置了此选项时，**multipathd** 守护程序才会检查新发现的路径或重新启用的路径的永久保留。

您可以将该属性添加到 multipath.conf 的 defaults 或 multipaths 部分。例如：

```
multipaths {
    multipath {
        wwid             3600140508dbcf02acb448188d73ec97d
        alias            yellow
        reservation_key  0x123abc
    }
}
```

为适用于永久管理的所有 mpath 设备设置 reservation_key 参数后，使用 **multipathd reconfigure** 重新装载配置。

注意：使用“reservation_key file”

如果在 multipath.conf 的 defaults 部分使用了特殊值 reservation_key file，则可以在文件 /etc/multipath/prkeys 中使用 **mpathpersist** 动态管理保留密钥。

这是处理多路径映射的永久保留的建议方法。从 SUSE Linux Enterprise Server 12 SP4 开始可以使用这种方法。

使用命令 **mpathpersist** 查询和设置由 SCSI 设备组成的多路径映射的永久保留。有关细节，请参见手册页 mpathpersist(8)。命令行选项与 sg3_utils 软件包中 **sg_persist** 的选项相同。sg_persist(8) 手册页详细解释了选项的语义。

在以下示例中，DEVICE 表示设备映射程序多路径设备，例如 /dev/mapper/mpatha。以下命令连同长选项一起列出，以便于阅读。所有选项都可以替换为单个字母，例如 **mpathpersist -oGS 123abc** DEVICE。

mpathpersist --in --read-keys DEVICE
: 读取设备的已注册保留密钥。

mpathpersist --in --read-reservation DEVICE
: 显示设备的现有保留。

mpathpersist --out --register --param-sark=123abc DEVICE
: 为设备注册一个保留密钥。这会为主机上的所有 I_T 节点（路径设备）添加保留密钥。

mpathpersist --out --reserve --param-rk=123abc --prout-type=5 DEVICE
: 使用先前注册的密钥为设备创建类型 5（“独占写入 - 仅限注册者”）保留。

mpathpersist --out --release --param-rk=123abc --prout-type=5 DEVICE
: 释放设备的类型 5 保留。

mpathpersist --out --register-ignore --param-sark=0 DEVICE
: 从设备中删除现有的保留密钥。

18.5 针对多路径配置系统

18.5.1 启用、启动和停止多路径服务

要允许多路径服务在引导时启动，请运行以下命令：

```
> sudo systemctl enable multipathd
```

要在正在运行的系统中手动启动该服务，请输入：

```
> sudo systemctl start multipathd
```

要重启动该服务，请输入：

```
> sudo systemctl restart multipathd
```

在大多数情况下不需要重启动该服务。要简单地让 **`multipathd`** 重新装载其配置，请运行：

```
> sudo systemctl reload multipathd
```

要检查该服务的状态，请输入：

```
> sudo systemctl status multipathd
```

要停止当前会话中的多路径服务，请运行：

```
> sudo systemctl stop multipathd
> sudo systemctl stop multipathd.socket
```

警告：禁用 multipathd

强烈建议始终在有权访问多路径硬件的每台主机上启用并运行 `multipathd.service`。但是，有时可能因为去除了多路径硬件、要部署其他一些多路径软件或出于查错目的，而需要禁用该服务。

要**仅为单次系统引导**禁用多路径，请使用内核参数 `multipath=off`。这会影响引导的系统和初始 ramfs（在这种情况下不需要重构建）。

要**永久**禁用 multipathd 服务，使其不会在今后引导系统时启动，请运行以下命令：

```
> sudo systemctl disable multipathd
> sudo systemctl disable multipathd.socket
> sudo dracut --force --omit multipath
```

（每当禁用或启用多路径服务后，都请重构建 `initrd`。请参阅第 18.3.2.3 节 “使初始 RAM 磁盘保持同步”。）

（可选）此外，如果您还想确保不设置多路径设备（即使是手动运行 **`multipath`** 时），请在重构建 initrd 之前，将以下几行添加到 `/etc/multipath.conf` 的末尾：

```
blacklist {
    wwid .*
}
```

18.5.2 针对多路径准备 SAN 设备

配置 SAN 设备的多路径 I/O 之前，请按需要执行以下操作准备 SAN 设备：

- 使用供应商工具配置 SAN 并设置区域。
- 使用供应商工具为储存阵列上的主机 LUN 配置访问权限。
- 如果 SUSE Linux Enterprise Server 未随附主机总线适配器 (HBA) 的驱动程序，请安装 HBA 供应商提供的 Linux 驱动程序。有关更多细节，请参见供应商的特定说明。

如果已检测到多路径设备并已启用 `multipathd.service`，则应会自动创建多路径映射。如果未自动创建，请使用 **`lsscsi`** 之类的命令检查低级设备的探测。另外，请使用 **`journalctl -b`** 检查系统日志。如果 HBA 驱动程序未检测到这些 LUN，请检查 SAN 中的区域设置。特别是要检查 LUN 屏蔽是否是活动的，以及是否已将 LUN 正确指派给服务器。

如果 HBA 驱动程序可以检测到 LUN，但未创建相应的块设备，则可能需要使用额外的内核参数。请参见 SUSE 知识库 (https://www.suse.com/support/kb/doc.php?id=3955167) 中的 TID 3955167：Troubleshooting SCSI (LUN) Scanning Issues（SCSI (LUN) 扫描问题查错）。

18.5.3 多路径设备上的分区

多路径映射可以像其路径设备一样包含分区。分区表扫描以及为分区创建设备节点的操作是由 **kpartx** 工具在用户空间中执行的。**kpartx** 由 udev 规则自动调用；通常不需要手动运行它。从技术上讲，kpartx 创建的“分区”设备也是设备映射程序设备，这些设备只会映射来自父设备的线性范围的块。可以通过 `/dev/disk/by-id/dm-uuid-partN-mpath-WWID` 可靠地访问多路径设备的具有已知 WWID 的第 `N` 个分区。

注意：禁用 kpartx 调用

可以在 `/etc/multipath.conf` 中使用 `skip_kpartx` 选项来禁用对选定的多路径映射调用 **kpartx**。例如，在虚拟化主机上，这种做法可能很有用。

可以像平时一样使用 YaST 或者使用 **fdisk** 或 **parted** 等工具来操作多路径设备上的分区表和分区。当分区工具退出时，系统将会记下应用于分区表的更改。如果这种方法不起作用（通常是因为设备繁忙），请尝试运行 **multipathd reconfigure** 或重引导系统。

不能以其他方式使用已分区的多路径设备。例如，不能从已分区的设备创建 LVM 物理卷。在执行此操作之前，需要擦除分区表。

18.6 多路径配置

内置的 `multipath-tools` 默认值适用于大多数设置。如需进行自定义，需要创建一个配置文件。主配置文件为 `/etc/multipath.conf`。此外，与软件集 `/etc/multipath/conf.d/*.conf` 匹配的文件将按字母顺序读取。有关优先级规则，请参见第 18.6.2 节“multipath.conf 语法”。

注意： 生成的配置文件

文件 `/etc/multipath/wwids`、`/etc/multipath/bindings` 和 `/etc/multipath/prkeys` 由 `multipath-tools` 维护，分别用于储存有关先前创建的多路径映射、映射名称和 SCSI 永久保留的保留密钥的永久信息。请不要编辑生成的这些配置文件。

注意：可配置的路径

除 `/etc/multipath.conf` 之外的其他配置目录和文件的路径是可配置的，但强烈建议不要更改这些路径。

18.6.1 创建 /etc/multipath.conf 文件

您可以从内置默认值生成 `multipath.conf` 模板。这会将所有默认设置设为显式设置。除非修改生成的文件，否则 `multipath-tools` 的行为不会更改。要生成配置模板，请运行：

```
multipath -T >/etc/multipath.conf
```

或者，可以创建一个极简的 `/etc/multipath.conf`，其中仅包含您要更改的那些设置。该行为等同于仅修改生成的模板中的相应行。

18.6.2 multipath.conf 语法

`/etc/multipath.conf` 文件使用由部分、子部分和属性/值对组成的层次结构。

- 令牌由空格分隔。除非括在引号中（参见下文），否则连续的空格字符将折叠成一个空格。
- 使用井号 (`#`) 和感叹号 (`!`) 字符会导致将行中的其余内容作为注释丢弃。
- 部分和子部分在同一行中以部分名称和左大括号 (`{`) 开头，并在独立的一行中以右大括号 (`}`) 结尾。
- 属性和值编写在一行中。不支持续行。
- 属性和部分名称必须是关键字。`multipath.conf(5)` 中阐述了允许的关键字。
- 值可以括在双引号 (`"`) 中。如果值包含空格或注释字符，则必须将其括在引号中。值中的双引号字符由一对双引号 (`""`) 表示。
- 某些属性的值是 POSIX 正则表达式（请参见 `regex(7)`）。它们区分大小写且位置不固定，例如，“`bar`”匹配“`rhabarber`”。

语法示例

```
section {
    subsection {
        attr1 value
    attr2      "complex value!"
 attr3    "value with ""quoted"" word"
    } ! subsection end
} # section end
```

优先级规则

如第 18.6 节 “多路径配置” 的开头所述，可以创建多个配置文件。附加的文件遵循与 `/etc/multipath.conf` 相同的语法规则。部分和属性可以多次出现。如果在多个文件中或者在同一文件的多行中设置了同一个属性，则读取的最后一个值优先。

18.6.3 `/etc/multipath.conf` 部分

`/etc/multipath.conf` 文件由下列部分构成。某些属性可以出现在多个部分中。有关详细信息，请参见 `multipath.conf(5)`。

defaults

一般默认设置。

blacklist

列出要忽略的设备。请参见第 18.8 节 “将非多路径设备加入黑名单” 。

blacklist_exceptions

列出要进行多路径处理的设备，即使它们与黑名单匹配。请参见第 18.8 节 “将非多路径设备加入黑名单” 。

devices

特定于储存控制器的设置。此部分是 `device` 子部分的集合。此部分中的值会覆盖 `defaults` 部分中相同属性的值。

multipaths

单个多路径设备的设置。此部分是 `multipath` 子部分的列表。值将覆盖 `defaults` 和 `devices` 部分。

overrides

覆盖所有其他部分中的值的设置。

18.6.4 应用 /etc/multipath.conf 修改

要应用配置更改，请运行

```
> sudo multipathd reconfigure
```

请不要忘记与 initrd 中的配置同步。请参见第 18.3.2.3 节 “使初始 RAM 磁盘保持同步”。

警告：不要使用 **multipath** 应用设置

当 **multipathd** 正在运行时，请不要使用 **multipath** 命令应用新设置。否则可能导致设置不一致甚至损坏。

注意：校验已修改的设置

可以在应用已修改的设置之前先对其进行测试，方法是运行：

```
multipath -d -v2
```

此命令显示要使用建议的拓扑创建的新映射。但是，该命令不显示是否会去除/刷新映射。要获取更多信息，请运行以下命令：

```
multipath -d -v3 2>&1 | less
```

18.6.5 生成 WWID

为了在不同的路径上识别设备，多路径使用每台设备的全球通用标识 (WWID)。如果两个设备路径的 WWID 相同，则假定它们代表同一台设备。除非有绝对的理由，否则建议不要更改生成 WWID 的方法。有关细节，请参见 **`man multipath.conf`**。

18.7 配置巡回检测、排队及故障回复的策略

本节介绍用于实现容错的最重要 `multipath-tools` 配置参数。

polling_interval

对路径设备进行状态检查的时间间隔（以秒为单位）。默认值为 5 秒。将按此时间间隔检查有故障的设备。对于正常的设备，最多可将时间间隔增大至 `max_polling_interval` 秒。

no_path_retry

确定当给定多路径映射的所有路径都发生故障或消失时会发生什么情况。可能的值有：

fail

使多路径映射上的 I/O 失败。这会导致上层（例如挂载的文件系统）发生 I/O 错误。受影响的文件系统（也可能是整个主机）将进入降级模式。

queue

多路径映射上的 I/O 在设备映射程序层中排队，并在路径设备重新可用时发送到设备。这是避免丢失数据的最安全选项，但如果路径设备长时间不能恢复，则使用该值可能会造成负面影响。从设备读取数据的进程将会挂起并处于不间断休眠 (D) 状态。排队的数据会占用内存，而被占用的内存不可供进程使用。最终，内存将会耗尽。

`N`

`N` 是一个正整数。使映射设备保持排队模式 `N` 个巡回检测间隔。在这段时间消逝后，**`multipathd`** 将使映射设备失败。如果 `polling_interval` 为 5 秒且 `no_path_retry` 为 6，则 **`multipathd`** 会将 I/O 排队大约 6 * 5 = 30 秒，然后使映射设备上的 I/O 失败。精心选择超时值通常可以在使 `I/O` 失败与排队之间做出合理的权衡。

多路径 I/O 旨在于储存系统与服务器之间提供连接容错。所需的默认行为取决于服务器是独立服务器还是高可用性群集中的一个节点。

在配置独立服务器的多路径 I/O 时，`no_path_retry` 设置会保护服务器操作系统在尽可能长的时间内不会接收 I/O 错误。它会在出现多路径故障转移前将消息排队，并提供正常的连接。

在为高可用性群集中的节点配置多路径 I/O 时，您需要让多路径报告 I/O 故障，以触发资源故障转移而不是等待多路径故障转移被解决。在群集环境中，您必须修改 `no_path_retry` 设置，以确保当与储存系统断开连接时，群集节点会收到与群集验证进程相关的 I/O 错误（建议为 Heartbeat 容错的 50%）。此外，您还希望将多路径 I/O 错误回复设为手动，以避免因路径故障而造成资源的乒乓效应。

`/etc/multipath.conf` 文件应包含一个 **`defaults`** 部分，您可以在此处指定巡回检测、排队以及故障回复的默认行为。如果该字段未另在 **`device`** 部分中指定，则为该 SAN 配置应用默认设置。

以下各项是已编译的默认设置。除非在创建及配置个性化 `/etc/multipath.conf` 文件时被重写，否则将使用这些值。

```
defaults {
  verbosity 2
#  udev_dir is deprecated in SLES 11 SP3
#  udev_dir              /dev
  polling_interval      5
#  path_selector default value is service-time in SLES 11 SP3
#  path_selector         "round-robin 0"
  path selector         "service-time 0"
  path_grouping_policy  failover
#  getuid_callout is deprecated in SLES 11 SP3 and replaced with uid_attribute
#  getuid_callout        "/usr/lib/udev/scsi_id --whitelisted --device=/dev/%n"
#  uid_attribute is new in SLES 11 SP3
  uid_attribute         "ID_SERIAL"
  prio                  "const"
  prio_args             ""
  features              "0"
  path_checker          "tur"
  alias_prefix          "mpath"
  rr_min_io_rq          1
  max_fds               "max"
```

```
rr_weight              "uniform"
queue_without_daemon   "yes"
flush_on_last_del      "no"
user_friendly_names    "no"
fast_io_fail_tmo       5
bindings_file          "/etc/multipath/bindings"
wwids_file             "/etc/multipath/wwids"
log_checker_err        "always"

retain_attached_hw_handler  "no"
detect_prio            "no"
failback               "manual"
no_path_retry          "fail"
}
```

有关巡回检测、排队和故障回复策略设置的信息，请参见第 18.10 节 “配置路径故障转移策略和优先级” 中的以下参数：

- polling_interval
- no_path_retry
- failback

修改 /etc/multipath.conf 文件后，必须在系统上运行 **dracut** -f 重新创建 initrd，然后重启动服务器，更改才会生效。有关详细信息，请参见 第 18.6.4 节 “应用 /etc/multipath.conf 修改” 。

18.8 将非多路径设备加入黑名单

/etc/multipath.conf 文件可能包含 **blacklist** 部分，其中会列出所有非多路径设备。您可以通过 WWID（wwid 关键词）、设备名称（devnode 关键词）或设备类型（device 部分）将设备加入黑名单。您还可以使用 blacklist_exceptions 部分，对通过 blacklist 部分中所用的正规表达式加入黑名单的一些设备启用多路径。

注意：首选的加入黑名单方法

将设备加入黑名单的首选方法是通过 **WWID** 或通过**供应商和产品**进行。建议不要通过 **devnode** 加入黑名单，因为设备节点可能会更改，因此对长久标识设备无帮助。

警告：multipath.conf 中的正则表达式

`/etc/multipath.conf` 中的正则表达式一般情况下**不**起作用。只有与通用字符串匹配时，它们才会起作用。不过，多路径的标准配置已包含许多设备和供应商的正则表达式。将正则表达式与其他正则表达式相匹配不起作用。请务必只与运行 **`multipath -t`** 后显示的字符串进行匹配。

您通常可能会忽略非多路径设备，例如 `hpsa`、`fd`、`hd`、`md`、`dm`、`sr`、`scd`、`st`、`ram`、`raw` 和 `loop`。例如，本地 SATA 硬盘和闪存盘没有多个路径。如果您希望 **`multipath`** 忽略单路径设备，请将它们放在 **`blacklist`** 部分。

注意：兼容性

关键词 `devnode_blacklist` 已弃用且已替换为 `blacklist`。

在 SUSE Linux Enterprise Server 12 中使用 glibc 提供的正规表达式。要与任意字符串匹配，现在必须使用 `".*"` 而不应使用 `"*"`。

例如，要将本地设备和 `hpsa` 驱动程序中的所有阵列加入黑名单，以免受到多路径的管理，则 **`blacklist`** 部分应为：

```
blacklist {
      wwid "26353900f02796769"
      devnode "^(ram|raw|loop|fd|md|dm-|sr|scd|st)[0-9]*"
      devnode "^sd[a-z][0-9]*"
}
```

您可以仅将一个驱动程序中的分区而不是整个阵列加入黑名单。例如，您可以使用以下正规表达式，只将 cciss 驱动程序中的分区而非整个阵列加入黑名单：

```
blacklist {
```

```
      devnode "^cciss!c[0-9]d[0-9]*[p[0-9]*]"
}
```

您可以在 blacklist 中添加 device 部分，并使用 vendor 和 product 关键词按特定设备类型加入黑名单。

```
blacklist {
      device {
           vendor  "DELL"
           product ".*"
       }
}
```

您可以使用 blacklist_exceptions 部分，对通过 blacklist 部分中所用的正规表达式加入黑名单的一些设备启用多路径。您可以通过 WWID（wwid 关键词）、设备名称（devnode 关键词）或设备类型（device 部分）添加例外。您必须以将相应设备加入黑名单的相同方式指定例外。也就是说，wwid 例外适用于 wwid 黑名单，devnode 例外适用于 devnode 黑名单，而设备类型例外适用于设备类型黑名单。

例如，如果您拥有来自不同供应商的设备类型，可以针对所需的设备类型启用多路径。在 blacklist 部分中将所有供应商的设备类型加入黑名单，然后在 blacklist_exceptions 部分中添加 device 部分，针对所需的设备类型启用多路径。

```
blacklist {
      devnode "^(ram|raw|loop|fd|md|dm-|sr|scd|st|sda)[0-9]*"
      device {
           vendor  "DELL"
           product ".*"
       }
}

blacklist_exceptions {
      device {
           vendor  "DELL"
           product "MD3220i"
       }
}
```

您还可以使用 blacklist_exceptions 仅针对特定设备启用多路径。例如：

```
blacklist {
      wwid ".*"
}

blacklist_exceptions {
        wwid "3600d0230000000000e13955cc3751234"
        wwid "3600d0230000000000e13955cc3751235"
}
```

修改 /etc/multipath.conf 文件后，必须在系统上运行 **dracut** -f 重新创建 initrd，然后重启动服务器，更改才会生效。有关详细信息，请参见 第 18.6.4 节 “应用 /etc/multipath.conf 修改”。

重引导之后，当您发出 **multipath -ll** 命令时，这些本地设备应不再列于多路径映射中。

注意：使用 find_multipaths 选项

从 SUSE Linux Enterprise Server 12 SP2 开始，多路径工具支持 /etc/multipath.conf 的 defaults 部分中的 find_multipaths 选项。此选项会阻止多路径和 multipathd 为只有单个路径的设备设置多路径映射（有关细节，请参见 **man 5 multipath.conf**）。如此，在某些配置中，管理员就无需为本地 SATA 磁盘等创建黑名单项。

虽然使用 find_multipaths 选项看似很方便，但也有其弊端。它使得系统引导复杂化且速度变慢，因为对于找到的每台设备，引导逻辑都需要等到发现了所有设备之后才能确定设备是否存在第二个路径。此外，当一些路径已关闭或在引导期间不可视时可能会出现问题 — 可能会错误地将设备检测为单路径设备并激活，导致后面添加更多路径的操作失败。

find_multipaths 会将 /etc/multipath/wwids 中所列的具有匹配 WWID 的所有设备视为多路径设备。请注意，第一次激活 find_multipaths 时不会产生任何效果，除非删除或编辑了 /etc/multipath/wwids，因为所有先前存在的多路径映射（包括单路径映射）都列于该 wwids 文件中。在具有多路径根文件系统的 SAN-boot 系统中，请确保 /etc/multipath/wwids 在初始 RAM 磁盘和文件系统之间保持同步。

总而言之，虽然在某些情况下使用 find_multipaths 可能很方便，但 SUSE 仍建议使用正确配置了黑名单和黑名单例外的默认配置。

18.9 配置用户用好的名称或别名

多路径设备可以按 WWID、用户友好的名称或您指派的别名进行标识。格式为 /dev/sdn 和 /dev/dm-n 的设备节点名称在重引导时会发生更改，且每次都可能会被指派给不同的设备。而设备的 WWID、用户友好名称以及别名则在重引导期间始终不变，因此是标识设备的首选方式。

重要：建议使用永久的名称

因为格式为 /dev/sdn 和 /dev/dm-n 的设备节点名称在重引导时会发生更改，所以最好使用其 WWID 来表示多路径设备。为了在重引导时唯一标识设备，您也可以使用与 WWID 映射的用户友好名称或别名。

下表介绍了 /etc/multipath.conf 文件中可用于设备的设备名称类型。有关 multipath.conf 设置的示例，请参见 /usr/share/doc/packages/multipath-tools/multipath.conf.synthetic 文件。

表 18.1： 多路径设备名称类型比较

名称类型	说明
WWID（默认）	WWID（全球标识符）序列是保证全球唯一且不会更改的多路径设备标识符。在多路径中使用的默认名称是 /dev/disk/by-id 目录中的逻辑单元的 ID。例如，WWID 为 3600508e0000000009e6baa6f609e7908 的设备将作为 /dev/disk/by-id/scsi-3600508e0000000009e6baa6f609e7908 列出。
用户友好	/dev/mapper 目录中的设备映射程序多路径设备名称也参考逻辑单元的 ID。这些多路径设备名称都是用户友好的名称，格式为 /dev/mapper/mpath<N>，比如 /dev/mapper/mpath0。名称是唯一且永久的，因为它们使用 /var/lib/multipath/bindings 文件跟踪 UUID 和用户友好的名称之间的关联。
Alias	别名是由管理员为多路径设备提供的全局唯一名称。别名会覆盖 WWID 和用户友好的 /dev/mapper/mpathN 名称。

名称类型	说明
	如果您使用的是 user_friendly_names，请不要将别名设为 mpathN 格式。否则可能会与自动指派的用户友好名称产生冲突，导致所提供的设备节点名称不正确。

/etc/multipath.conf 文件中的全局多路径 user_friendly_names 选项用于对多路径设备启用或禁用用户友好的名称。如果将它设置为 no（默认值），多路径会使用 WWID 作为设备的名称。如果将它设置为 yes，多路径将使用 /var/lib/multipath/bindings 文件，在 /dev/mapper 目录中以 mpath<N> 格式为设备指派一个永久的唯一名称。/etc/multipath.conf 文件中的 bindings_file 选项可用于指定 bindings 文件的备用位置。

/etc/multipath.conf 文件中的全局多路径 alias 选项用于明确为设备指派一个名称。如果为多路径设备设置了别名，则使用别名，而不使用 WWID 或用户友好的名称。

使用 user_friendly_names 选项时，在以下情形中可能会出现问题：

根设备正在使用多路径：

如果系统根设备使用的是多路径，而您使用的是 user_friendly_names 选项，则 initrd 文件中将会包含 /var/lib/multipath/bindings 文件中的用户友好设置。如果稍后更改储存设置，比如添加或删除了设备，则 initrd 内的绑定设置和 /var/lib/multipath/bindings 中的绑定设置之间会存在不匹配。

警告：绑定不匹配项

initrd 和 /var/lib/multipath/bindings 之间的绑定不匹配会导致对设备指派错误的挂载点，进而导致文件系统损坏和数据丢失。

为了避免此问题，建议对系统根设备使用默认 WWID 设置。对于系统根设备不应使用别名。由于设备名称不同，因此使用别名会导致您无法通过内核命令行无缝关闭多路径。

从另一个分区挂载 /var：

user_friendly_names 配置文件的默认位置是 /var/lib/multipath/bindings。如果 /var 数据不是位于系统 root 设备上，而是从另一个分区挂载，则设置多路径时 bindings 文件不可用。

请确保 /var/lib/multipath/bindings 文件在系统 root 设备上可用，并且多路径可以找到它。例如，可以按如下操作来确保这一点：

1. 将 /var/lib/multipath/bindings 文件移动到 /etc/multipath/bindings。
2. 将 /etc/multipath.conf 的 defaults 部分的 bindings_file 选项设置为此新位置。例如：

```
defaults {
                user_friendly_names yes
                bindings_file "/etc/multipath/bindings"
}
```

initrd 中存在多路径：

即使系统根设备不在多路径上，多路径也可以包含在 initrd 中。例如，如果系统根设备位于 LVM 上，就可能会发生这种情况。如果您使用 user_friendly_names 选项，且多路径在 initrd 中，则应当使用参数 **multipath=off** 进行引导，以免出现问题。这样在系统引导过程中仅禁用了 initrd 中的多路径。系统引导后，boot.multipath 和 multipathd 引导脚本能够激活多路径。

HA 群集中的多路径：

有关详细信息，请参见 第 18.9.1 节 “HA 群集中的多路径设备名称”。

启用用户友好名称或指定别名：

1. 使用 root 权限在文本编辑器中打开 /etc/multipath.conf 文件。
2. （可选）修改 /var/lib/multipath/bindings 文件的位置。

   备用路径必须在系统根设备上可用，且多路径可以找到它。

   a. 将 /var/lib/multipath/bindings 文件移动到 /etc/multipath/bindings。
   b. 将 /etc/multipath.conf 的 defaults 部分的 bindings_file 选项设置为此新位置。例如：

```
defaults {
```

```
            user_friendly_names yes
            bindings_file "/etc/multipath/bindings"
}
```

3. （可选但不建议使用）启用用户友好的名称：

   a. 取消注释 `defaults` 部分及其结尾括号。

   b. 取消注释 `user_friendly_names` 选项，然后将其值从 No 更改为 Yes。
   例如：

   ```
   ## Use user-friendly names, instead of using WWIDs as names.
   defaults {
     user_friendly_names yes
   }
   ```

4. （可选）使用 **`multipath`** 部分中的 **`alias`** 选项为设备指定您自己的名称。
   例如：

```
## Use alias names, instead of using WWIDs as names.
multipaths {
       multipath {
               wwid           360060480000283501312535 94d303030
               alias             blue1
       }
       multipath {
               wwid           360060480000283501312535 94d303041
               alias             blue2
       }
       multipath {
               wwid           360060480000283501312535 94d303145
               alias             yellow1
       }
       multipath {
               wwid           360060480000283501312535 94d303334
               alias             yellow2
       }
}
```

> **重要：WWID 与 WWN 的对比**
>
> 在 `/etc/multipath.conf` 文件中定义设备别名时，请确保使用每个设备的 WWID（如 `3600508e0000000009e6baa6f609e7908`）而不是它的 WWN（以 `0x` 替换设备 ID 的第一个字符，如 `0x600508e0000000009e6baa6f609e7908`）。

5. 保存更改，然后关闭该文件。
6. 修改 `/etc/multipath.conf` 文件后，必须在系统上运行 **`dracut`** `-f` 重新创建 `initrd`，然后重启动服务器，更改才会生效。有关详细信息，请参见 第 18.6.4 节 “应用 `/etc/multipath.conf` 修改”。

如果要直接使用整个 LUN（例如，如果您使用 SAN 功能对储存区分区），可为 **`mkfs`**、`/etc/fstab`、应用程序等使用 `/dev/disk/by-id/xxx` 名称。已分区设备的设备名称后将追加 `_part<n>`，例如 `/dev/disk/by-id/xxx_part1`。

在 `/dev/disk/by-id` 目录中，通过设备名 `dm-uuid*` 或别名（如果在 `/etc/multipath.conf` 文件中为其指派了别名）表示多路径映射设备。设备名 `scsi-` 和 `wwn-` 表示设备的物理路径。

18.9.1 HA 群集中的多路径设备名称

请执行下列操作来确保多路径设备名称在所有设备中都相同：

- 使用 UUID 和别名确保多路径设备名称在群集中的所有节点上保持一致。别名在所有节点上必须唯一。将 `/etc/multipath.conf` 文件从该节点复制到群集中所有其他节点的 `/etc/` 目录下。
- 当使用多路径映射设备的链接时，请确保在 `/dev/disk/by-id` 目录中指定 `dm-uuid*` 名称或别名，而不是设备的固定路径实例。有关信息，请参见第 18.9 节 “配置用户用好的名称或别名”。
- 将 `user_friendly_names` 配置选项设置为“no”以禁用它。用户友好名称对某个节点是唯一的，但群集中各个节点上的设备不得指派相同的用户友好名称。

注意：用户友好名称

如果您确实需要使用用户友好名称，则可以执行下列操作，强制系统定义的用户友好名称在群集的所有节点中保持一致：

1. 在一个节点上的 `/etc/multipath.conf` 文件中：

    1. 将 `user_friendly_names` 配置选项设置为“yes”以启用它。
       多路径会使用 `/var/lib/multipath/bindings` 文件，在 `/dev/mapper` 目录中以 `mpath<N>` 格式为设备指派一个永久的唯一名称。

    2. （可选）设置 `/etc/multipath.conf` 文件的 `defaults` 部分中的 `bindings_file` 选项，以指定 `bindings` 文件的备用位置。
       默认位置为 `/var/lib/multipath/bindings`。

2. 设置该节点上的所有多路径设备。

3. 将 `/etc/multipath.conf` 文件从该节点复制到群集中所有其他节点的 `/etc/` 目录下。

4. 从该节点将 `bindings` 文件复制到群集中所有其他节点的 `bindings_file` 路径。

5. 修改 `/etc/multipath.conf` 文件后，必须在系统上运行 **`dracut`** `-f` 重新创建 `initrd`，然后重启动节点，更改才会生效。有关详细信息，请参见 第 18.6.4 节“应用 `/etc/multipath.conf` 修改”。这适用于所有受影响的节点。

18.10 配置路径故障转移策略和优先级

在 Linux 主机中，当存在多个路径到某储存控制器时，每个路径都显示为独立的块设备，导致单个 LUN 有多个块设备。设备映射程序多路径服务检测到具有同一 LUN ID 的多个路径，并使用该 ID 创建了一个新的多路径设备。例如，如果具有两个 HBA 的主机通过单个未确定区域的

光纤通道交换机挂接到一个带两个端口的储存控器，则会看到四个块设备：/dev/sda、/dev/sdb、/dev/sdc 和 /dev/sdd。设备映射程序多路径服务会创建单个块设备 /dev/mpath/mpath1，由它来通过以上四个底层块设备重新路由 I/O。

本节描述如何指定故障转移策略和配置路径的优先级。请注意，修改 /etc/multipath.conf 文件后，必须在系统上运行 **dracut** -f 重新创建 initrd，然后重启动服务器，更改才会生效。有关详细信息，请参见 第 18.6.4 节 “应用 /etc/multipath.conf 修改”。

18.10.1 配置路径故障转移策略

使用带 -p 选项的 **multipath** 命令来设置路径故障转移策略：

```
> sudo multipath DEVICENAME -p POLICY
```

将 POLICY 替换为以下策略选项之一：

表 18.2： MULTIPATH -P 命令的组策略选项

策略选项	说明
failover	（默认）每个优先级组对应一个路径。
multibus	一个优先级组中的所有路径。
group_by_serial	每个检测到的序列号一个优先级组。
group_by_prio	每个路径优先级值一个优先级组。优先级由调出程序在 /etc/multipath.conf 配置文件中指定为 global、per-controller 或 per-multipath 选项来确定。
group_by_node_name	每个目标节点名一个优先级组。目标节点名是在 /sys/class/fc_transport/target*/node_name 位置获取的。

18.10.2 配置故障转移优先级

必须在 `/etc/multipath.conf` 文件中手动输入设备的故障转移优先级。`/usr/share/doc/packages/multipath-tools/multipath.conf.annotated` 文件中有所有设置和选项的示例。

18.10.2.1 了解优先级组和属性

优先级组是通向同一物理 LUN 的路径的集合。默认情况下，I/O 在组中的所有路径中以循环方式分布。**`multipath`** 命令会基于 SAN 的 path_grouping_policy 设置为该 SAN 中的每个 LUN 自动创建优先级组。**`multipath`** 命令将一个组中的路径数乘以该组的优先级以确定哪个组是主组。计算出的值最高的组即是主组。当主组中的所有路径失败时，具有下一个最高值的优先级组即变为活动的组。

路径优先级是指派给一个路径的整数值。值越高，优先级越高。将使用一个外部程序来指派每个路径的优先级。对于给定的设备，具有相同优先级的路径属于同一优先级组。

`prio` 设置在 `/etc/multipath.conf` 文件的 `defaults{}` 或 `devices{}` 部分中使用。当为 `multipaths` 部分的单个 `multipath` 定义指定该设置时，系统会无提示地将它忽略。`prio` 行用于指定优先级排序程序。如果优先级排序程序需要参数，您要在第二行上使用 `prio_args` 关键字指定参数。

Defaults 或 Devices 部分的 PRIO 设置

`prio`

指定要调用的优先级排序程序，以获取路径优先级的值。将加总每个路径组的权重以确定发生故障时要使用的下一个路径组。

如果指定的优先级排序程序需要参数，则使用关键字 `prio_args` 指定参数。

如果未指定关键字 `prio`，则所有路径具有相同优先级。默认设置为 `const`，且 `prio_args` 设置为空。

```
prio "const"
prio_args ""
```

优先级排序程序示例如下：

优先级排序程序	说明
alua	基于 SCSI-3 ALUA 设置生成路径优先级。
const	为所有路径生成同一优先级。
emc	为 EMC 阵列生成路径优先级。
hdc	为 Hitachi HDS Modular 储存阵列生成路径优先级。
hp_sw	为活动/备用模式的 Compaq/HP 控制器生成路径优先级。
ontap	为 NetApp 阵列生成路径优先级。
random	为每个路径生成随机优先级。
rdac	为 LSI/Engenio RDAC 控制器生成路径优先级。
weightedpath	根据您在 prio_args 的参数中指定的加权值生成路径优先级。
path_latency	根据延迟算法生成路径优先级，该算法通过 prio_args 关键字配置。

prio_args 参数

这些是需要参数的优先级排序程序的参数。大多数 prio 程序并不需要参数。无默认值。值取决于 prio 设置以及优先级排序程序是否需要以下任何参数：

weighted

需要格式为 [hbtl|devname|serial|wwn] REGEX1 PRIO1 REGEX2 PRIO2... 的值正则表达式必须采用 SCSI H:B:T:L 格式，例如 1:0:.:.和 *:0:0:.，并带有加权值，其中 H、B、T 和 L 分别代表设备的主机、总线、目标以及 LUN ID。例如：

```
prio "weightedpath"
prio_args "hbtl 1:.:.:. 2 4:.:.:. 4"
```

devname

Regex 为设备名称格式。例如：sda、sd.e

serial

Regex 为序列号格式。例如：.*J1FR.*324。使用 **`multipathd show paths format %z`** 命令查询您的序列号。（**`multipathd show wildcards`** 会显示所有 `format` 通配符。）

alua

如果为某个设备 (`alua exclusive_pref_bit`) 设置了 `exclusive_pref_bit`，那么设置了 `preferred path` 的路径将始终在其自身的路径组中。

path_latency

如果远程储存阵列和本地储存阵列使用的硬件类型相同，则可以使用 `path_latency` 来调整这两个阵列之间的延迟。远程阵列上的延迟通常更高，因此您可以调整延迟，使两个阵列的时间更接近。这需要格式为 `io_num=20 base_num=10` 的值对。

`io_num` 是连续发送到当前路径的读取 IO 数，用于计算平均路径延迟。有效值为 2 到 200 的整数。

`base_num` 是对数底数，用于将不同的优先级排名分区。有效值为 2 到 10 的整数。最大平均延迟值为 100s，最小值为 1 µs。例如，如果 `base_num=10`，则路径将被分到路径延迟 <=1 µs、(1 µs, 10 µs]、(10 µs, 100 µs)、(100 µs, 1 ms)、(1 ms, 10 ms)、(10 ms, 100 ms)、(100 ms, 1 s)、(1 s, 10 s)、(10 s, 100 s)、>100 s 的各优先级组。

多路径属性

多路径属性用于控制设备的多路径 I/O 行为。您可以按默认方式为所有多路径设备指定属性。也可以通过在多路径配置文件的 `multipaths` 部分中创建设备项的方式，指定仅对指定多路径设备应用的属性。

`user_friendly_names`

指定是使用全球 ID (WWID) 还是使用 `/var/lib/multipath/bindings` 文件以 `/dev/mapper/mpathN` 的格式为多路径设备指派一个永久唯一的别名。

此选项可以在 `devices` 部分和 `multipaths` 部分使用。

值	说明
no	（默认）使用 /dev/disk/by-id/ 位置中显示的 WWID。
yes	自动生成用户友好的名称作为多路径设备的别名而不是实际 ID。

failback

指定是否监视失败路径恢复，并指明失败路径重新使用之后组故障回复的时间。

当失败的路径恢复后，系统会根据此设置将该路径重新添加到支持多路径的路径列表中。多路径计算优先级组，并在主路径的优先级超出备用组时更改活动优先级组。

值	说明
manual	（默认）不监视失败的路径是否恢复。管理员运行 **multipath** 命令以更新启用的路径和优先级组。
followover	仅当路径组的第一个路径激活时才执行自动故障回复。这可以防止某个节点在另一个节点请求了故障转移时自动故障回复。
immediate	当一个路径恢复时，立即启用该路径。
N	当路径恢复时，等待 N 秒再启用该路径。指定大于 0 的整数值。

为了防止多路径故障转移出现乒乓效应，建议将群集环境下的多路径故障回复设置为 manual。

```
failback "manual"
```

重要：校验

请务必向您的储存系统供应商确认故障回复设置。不同的储存系统可能会有不同的设置要求。

no_path_retry

指定在路径失败时使用的行为。

值	说明
N	指定 **multipath** 停止排队和路径失败之前的重试次数。指定大于 0 的整数值。 在群集中，您可以将值指定为 0 以避免发生排队现象，并允许资源进行故障转移。
fail	指定立即失败（不排队）。
queue	永不停止排队（一直排队，直到该路径成为活动路径）。

在群集中工作时，建议在 /etc/multipath.conf 文件中将重试设置设为 fail 或 0。如此一来，当与储存的连接中断时便可让资源进行故障转移。否则会将消息排队而不会发生资源故障转移。

```
no_path_retry "fail"
no_path_retry "0"
```

重要：校验

请务必向您的储存系统供应商确认重试设置。不同的储存系统可能会有不同的设置要求。

path_checker

确定路径的状态。

值	说明
directio	读取具有直接 I/O 的第一个扇区，这对于 DASD 设备非常有用。在 systemd 日志中记录失败消息（请参见《管理指南》，第 21 章 “**journalctl**：查询 systemd 日志”）。

值	说明
`tur`	向设备发出 SCSI 测试单元就绪命令。如果 LUN 支持，则这是首选设置。如果命令失败，它不会在 `systemd` 日志中填入消息。
`CUSTOM_VENDOR_VALUE`	一些 SAN 供应商提供了自定义的 path_checker 选项： • `cciss_tur`： 检查 HP 智能储存阵列的路径状态。 • `emc_clariion`： 查询 EMC Clariion EVPD 页 0xC0 以确定路径状态。 • `hp_sw`： 使用活动/备用固件检查 HP 储存阵列的路径状态（Up、Down 还是 Ghost）。 • `rdac`： 检查 LSI/Engenio RDAC 储存控制器的路径状态。

`path_grouping_policy`

指定由给定控制器托管的多路径设备的路径分组策略。

值	说明
`failover`	（默认）为每个优先级组指派一个路径，以便每次只使用一个路径。
`multibus`	所有有效路径在一个优先级组中。通讯在该组所有活动路径之间平衡负载。
`group_by_prio`	对于每个路径优先级值，都存在一个优先级组。具有相同优先级的路径在同一优先级组中。优先级由外部程序指派。

值	说明
`group_by_serial`	路径按 SCSI 目标序列号（控制器节点 WWN）分组。
`group_by_node_name`	每个目标节点名指派一个优先级组。目标节点名是在 `/sys/class/fc_transport/target*/node_name` 中获取的。

`path_selector`

指定用于负载平衡的 path-selector 算法。

值	说明
`round-robin 0`	用于平衡优先级组中所有活动路径之间的通讯的负载平衡算法。
`queue-length 0`	用于平衡路径上运行中 I/O 数的动态负载平衡器，类似于 least-pending 选项。
`service-time 0`	（默认值）面向服务时间的、用于根据延迟情况来平衡多个路径上的 I/O 的负载平衡器。

pg_timeout

指定路径组超时处理。不能指定任何值，已设置了内部默认值。

`polling_interval`

指定某个路径检查周期结束到下一个路径检查周期开始之间的时间（以秒为单位）。
指定大于 0 的整数值。默认值为 5。请务必向您的储存系统供应商确认 polling_interval 设置。不同的储存系统可能会有不同的设置要求。

`rr_min_io_rq`

指定在使用基于请求的 device-mapper-multipath 切换到当前路径组中的下一个路径前，路由至某个路径的 I/O 请求的数量。
指定大于 0 的整数值。默认值为 1。

```
rr_min_io_rq "1"
```

`rr_weight`

指定用于路径的加权方法。

值	说明
`uniform`	（默认）所有路径具有相同的循环复用权重。
`priorities`	每个路径的权重通过路径的优先级乘以 rr_min_io_rq 设置来确定。

`uid_attribute`

提供唯一路径标识符的 udev 属性。默认值为 `ID_SERIAL`。

18.10.2.2 配置循环负载平衡

所有路径都是活动的。I/O 被配置为等候数秒或一定的 I/O 事务数，再移到序列中下一个打开的路径。

18.10.2.3 配置单路径故障转移

具有最高优先级（设置值为最低）的一个路径是活动的，可以进行通讯。其他路径可用于故障转移，但在发生故障转移前不会使用。

18.10.2.4 将 I/O 路径分组以进行循环负载平衡

具有相同优先级的多个路径分在活动的组中。当该组中的所有路径失败时，设备会故障转移到下一个最高优先级的组。该组中的所有路径以循环负载平衡的方式共享通讯负载。

18.10.3 报告目标路径组

使用 SCSI 报告目标端口组 (**`sg_rtpg(8)`**) 命令。有关信息，请参见 **`sg_rtpg(8)`** 的手册页。

18.11 配置根设备的多路径 I/O

SUSE Linux Enterprise Server 中的 /boot 和 /root 可以使用并支持设备映射程序多路径 I/O (DM-MPIO)。此外，YaST 安装程序中的 YaST 分区程序支持在安装时启用多路径。

18.11.1 安装时启用多路径 I/O

要在多路径设备上安装操作系统，安装时必须运行多路径软件。multipathd 守护程序在系统安装过程中不会自动激活。您可以使用 YaST 分区程序中的配置多路径选项来启动它。

18.11.1.1 安装时在主动/主动多路径储存 LUN 上启用多路径 I/O

1. 安装期间，在建议的分区屏幕上选择专家分区程序。
2. 选择硬盘主图标，单击配置按钮，然后选择配置多路径。
3. 启动多路径。
   YaST 会开始重扫描磁盘，并显示可用的多路径设备（如 /dev/disk/by-id/dm-uuid-mpath-3600a0b80000f4593000012ae4ab0ae65）。它是应做所有进一步处理的设备。
4. 单击下一步继续安装。

18.11.1.2 安装时在主动/被动多路径储存 LUN 上启用多路径 I/O

multipathd 守护程序在系统安装过程中不会自动激活。您可以使用 YaST 分区程序中的配置多路径选项来启动它。

在安装时为主动/被动多路径储存 LUN 启用多路径 I/O：

1. 安装期间，在建议的分区屏幕上选择专家分区程序。
2. 选择硬盘主图标，单击配置按钮，然后选择配置多路径。
3. 启动多路径。

YaST 会开始重扫描磁盘，并显示可用的多路径设备（如 /dev/disk/by-id/dm-uuid-mpath-3600a0b80000f4593000012ae4ab0ae65）。它是应做所有进一步处理的设备。记下设备路径和 UUID，您以后会需要这些信息。

4. 单击下一步继续安装。

5. 所有设置都设好且安装完成后，YaST 即会开始写入引导加载程序信息，并显示重启动系统的倒计时。单击停止按钮停止倒计时，然后按 Ctrl – Alt – F5 访问控制台。

6. 通过控制台确定是否在 /boot/grub/device.map 文件中为 hd0 条目输入了不活动路径。
   这一点是必需的，因为安装时不会区分活动和不活动路径。

   a. 输入以下命令将根设备挂载 /mnt

   ```
   > sudo mount /dev/disk/by-id/UUID;_part2 /mnt
   ```

   例如，输入

   ```
   > sudo mount /dev/disk/by-id/dm-uuid-
   mpath-3600a0b80000f4593000012ae4ab0ae65_part2 /mnt
   ```

   b. 输入以下命令将引导设备挂载 /mnt/boot

   ```
   > sudo mount /dev/disk/by-id/UUID_part1 /mnt/boot
   ```

   例如，输入

   ```
   > sudo mount /dev/disk/by-id/dm-uuid-
   mpath-3600a0b80000f4593000012ae4ab0ae65_part2 /mnt/boot
   ```

   c. 在 /mnt/boot/grub/device.map 文件中，确定 hd0 条目是否指向不活动路径，然后执行下列其中一项操作：

      - **活动路径：** 不需要执行任何操作。跳过所有剩余的步骤，按 Ctrl – Alt – F7 返回 YaST 图形环境，然后继续安装。
      - **不活动路径：** 必须更改配置并重新安装引导加载程序。

7. 如果 hd0 条目指向不活动路径，请更改配置并重新安装引导加载程序：

   a. 在控制台提示符处输入以下命令：

   ```
   mount -o bind /dev /mnt/dev
   mount -o bind /sys /mnt/sys
   mount -o bind /proc /mnt/proc
   chroot /mnt
   ```

   b. 在控制台中，运行 **multipath -ll**，然后检查输出以找到活动路径。
   不活动路径标有 ghost 字样。

   c. 在 /boot/grub/device.map 文件中，将 hd0 条目更改为活动路径，保存更改并关闭文件。

   d. 输入以下命令重新安装引导加载程序

   ```
   grub-install /dev/disk/by-id/UUID_part1 /mnt/boot
   ```

   例如，输入

   ```
   grub-install /dev/disk/by-id/dm-uuid-
   mpath-3600a0b80000f4593000012ae4ab0ae65_part2 /mnt/boot
   ```

   e. 输入以下命令：

   ```
   exit
   umount /mnt/*
   umount /mnt
   ```

8. 按 Ctrl – Alt – F7 返回 YaST 图形环境。

9. 单击确定继续安装重引导。

18.11.2 为现有根设备启用多路径 I/O

1. 将 Linux 安装为仅有一个路径活动，最好是分区程序中列出 by-id 符号链接的路径。

2. 使用安装期间使用的 /dev/disk/by-id 路径挂载设备。

3. 打开或创建 `/etc/dracut.conf.d/10-mp.conf`，并在其中添加以下行（请注意前导空格）：

```
force_drivers+=" dm-multipath"
```

4. 对于 IBM Z，请在运行 **`dracut`** 之前先编辑 `/etc/zipl.conf` 文件，以 `/etc/fstab` 中使用的相同 by-id 信息更改 `zipl.conf` 中的 by-path 信息。

5. 运行 **`dracut`** `-f` 更新 `initrd` 映像。

6. 对于 IBM Z，请在运行 **`dracut`** 之后运行 **`zipl`**。

7. 重引导服务器。

18.11.3 在根设备上禁用多路径 I/O

向内核命令行中添加 `multipath=off`。这可以通过 YaST 引导加载程序模块来完成。打开引导加载程序安装 › 内核参数，并将参数添加到两个命令行中。

这仅会影响根设备。不会影响所有其他设备。

18.12 为现有软件 RAID 配置多路径 I/O

理想的情况下，应先为设备配置多路径，再将这些设备用作软件 RAID 设备的组件。如果创建任何软件 RAID 设备之后添加多路径，则在重引导时，DM-MPIO 服务可能会比 **`multipath`** 服务启动得晚，导致多路径似乎不可用于 RAID。您可以使用本节中的过程使多路径对先前现有的软件 RAID 运行。

例如，您可能需要在以下情况中对软件 RAID 中的设备配置多路径：

- 如果在全新安装或升级期间将一个新的软件 RAID 创建为分区设置的一部分。
- 如果在软件 RAID 将设备用作成员设备或备用设备之前，没有将这些设备配置为支持多路径。
- 如果通过向服务器添加新的 HBA 适配器或扩展 SAN 中的储存子系统而增加了系统规模。

注意：假设条件

以下说明假定软件 RAID 设备是 `/dev/mapper/mpath0`，这是内核可以识别的设备名。它假设您已按照第 18.9 节 “配置用户用好的名称或别名” 所述在 `/etc/multipath.conf` 文件中启用了用户友好的名称。

确保将这些说明中的设备名修改为您的软件 RAID 的设备名。

1. 打开一个终端控制台。
   除非另行说明，请使用此控制台输入以下步骤中的命令。

2. 如果当前已挂载或正在运行任何软件 RAID 设备，请对每个设备输入以下命令来卸载并停止设备。

   ```
   > sudo umount /dev/mapper/mpath0
   > sudo mdadm --misc --stop /dev/mapper/mpath0
   ```

3. 输入以下命令停止 **md** 服务

   ```
   > sudo systemctl stop mdmonitor
   ```

4. 输入以下命令启动 `multipathd` 守护程序：

   ```
   > systemctl start multipathd
   ```

5. 启动多路径服务之后，校验软件 RAID 的组件设备是否列在 `/dev/disk/by-id` 目录中。
   执行以下操作之一：

   - **已列出设备：** 现在设备名应已具有到其设备映射程序多路径设备名的符号链接，例如 `/dev/dm-1`。

   - **未列出设备：** 输入以下命令清理并重新发现设备，强制多路径服务识别它们

     ```
     > sudo multipath -F
     > sudo multipath -v0
     ```

     这些设备现在应已在 `/dev/disk/by-id` 中列出，并且有到其设备映射程序多路径设备名的符号链接。例如：

```
lrwxrwxrwx 1 root root 10 2011-01-06 11:42 dm-uuid-
mpath-36006016088d014007e0d0d2213ecdf11 -> ../../dm-1
```

6. 输入以下命令重启动 `mdmonitor` 服务和 RAID 设备

```
> sudo systemctl start mdmonitor
```

7. 通过输入以下命令，检查软件 RAID 的状态

```
> sudo mdadm --detail /dev/mapper/mpath0
```

RAID 的组件设备应匹配在 `/dev/disk/by-id` 目录中列为设备符号链接的设备映射程序多路径设备名。

8. 如果根 (`/`) 设备或其任何部分（例如 `/var`、`/etc`、`/log`）位于 SAN 上，并且需要使用多路径引导，请重构建 `initrd`：

```
> dracut -f --add-multipath
```

9. 重引导服务器以应用更改。

10. 通过检查 RAID 状态，校验软件 RAID 阵列在多路径设备之上正确工作。输入

```
> sudo mdadm --detail /dev/mapper/mpath0
```

例如：

```
Number Major Minor RaidDevice State
0 253 0 0 active sync /dev/dm-0
1 253 1 1 active sync /dev/dm-1
2 253 2 2 active sync /dev/dm-2
```

注意：将 mdadm 与多路径设备配合使用

`mdadm` 工具要求使用 ID 而不是设备节点路径访问设备。有关细节，请参考第 18.4.3 节 “多路径设备上的 MD RAID”。

18.13 在多路径设备上使用 LVM2

使用多路径时，资源的所有路径都显示为设备树中的设备。默认情况下，LVM 会检查设备树中任何设备的顶层是否存在多路径设备。如果 LVM 在顶层找到了多路径设备，它会假设该设备是多路径组件，并忽略（底层）设备。这很可能是您期望的行为，但也可以在 `/etc/lvm/lvm.conf` 中更改。如果将 multipath_component_detection 设置为 1，LVM 会扫描多路径组件设备。lvm.conf 中的默认项是：

```
# By default, LVM2 will ignore devices used as component paths
# of device-mapper multipath devices.
# 1 enables; 0 disables.
multipath_component_detection = 1
```

18.14 最佳实践

18.14.1 在不重引导的情况下扫描新设备

如果已将系统配置为启用多路径，并且您稍后需要向 SAN 添加储存，则可以使用 **`rescan-scsi-bus.sh`** 脚本扫描新设备。默认情况下，此脚本将扫描具有典型 LUN 范围的所有 HBA。该命令的一般语法如下所示：

```
> sudo rescan-scsi-bus.sh [options] [host [host ...]]
```

对于大多数储存子系统，此脚本可在无选项的情况下成功运行。不过，在某些特殊情况下，可能需要使用一或多个选项。有关细节，请运行 **`rescan-scsi-bus.sh --help`**。

警告：EMC PowerPath 环境

在 EMC PowerPath 环境中，请勿使用操作系统随附的 `rescan-scsi-bus.sh` 实用程序或 HBA 供应商脚本来扫描 SCSI 总线。为了避免可能发生的文件系统损坏，EMC 要求您遵照 EMC PowerPath for Linux 的供应商文档中提供的过程操作。

使用以下过程扫描设备并在不重引导系统的情况下使这些设备可用于多路径。

1. 在储存子系统上，使用供应商的工具分配设备并更新其访问控制设置，以允许 Linux 系统访问新的储存区。有关细节，请参见供应商的文档。

2. 在所有目标中扫描查找一个主机，使其新设备为 Linux 内核 SCSI 子系统的中间层所知。在终端控制台提示符下，输入

```
> sudo rescan-scsi-bus.sh
```

根据您的设置，您可能需要结合可选参数运行 **rescan-scsi-bus.sh**。有关细节，请参见 **rescan-scsi-bus.sh --help**。

3. 请在 systemd 日志中检查扫描进度（有关细节，请参见《管理指南》, 第 21 章 “**journalctl**：查询 systemd 日志”）。在终端控制台提示符下，输入

```
> sudo journalctl -r
```

此命令显示日志的最后 行。例如：

```
> sudo journalctl -r
Feb 14 01:03 kernel: SCSI device sde: 81920000
Feb 14 01:03 kernel: SCSI device sdf: 81920000
Feb 14 01:03 multipathd: sde: path checker registered
Feb 14 01:03 multipathd: sdf: path checker registered
Feb 14 01:03 multipathd: mpath4: event checker started
Feb 14 01:03 multipathd: mpath5: event checker started
Feb 14 01:03:multipathd: mpath4: remaining active paths: 1
Feb 14 01:03 multipathd: mpath5: remaining active paths: 1
[...]
```

4. 重复上述步骤，以添加通过 Linux 系统上连接到新设备的其他 HBA 适配器的路径。

5. 运行 **multipath** 命令以识别 DM-MPIO 配置的设备。在终端控制台提示符下，输入

```
> sudo multipath
```

现在可以为多路径配置新设备了。

18.14.2 在不重引导的情况下扫描新的分区设备

使用本节中的示例，在不重引导的情况下检测新添加的多路径 LUN。

警告：EMC PowerPath 环境

在 EMC PowerPath 环境中，请勿使用操作系统随附的 `rescan-scsi-bus.sh` 实用程序或 HBA 供应商脚本来扫描 SCSI 总线。为了避免可能发生的文件系统损坏，EMC 要求您遵照 EMC PowerPath for Linux 的供应商文档中提供的过程操作。

1. 打开一个终端控制台。
2. 在所有目标中扫描查找一个主机，使其新设备为 Linux 内核 SCSI 子系统的中间层所知。在终端控制台提示符下，输入

   ```
   > rescan-scsi-bus.sh
   ```

   根据您的设置，您可能需要结合可选参数运行 **`rescan-scsi-bus.sh`**。有关细节，请参见 **`rescan-scsi-bus.sh --help`**。
3. 通过输入以下命令校验该设备是否可见（例如链接是否有一个新的时戳）

   ```
   > ls -lrt /dev/dm-*
   ```

   您也可以通过输入以下命令来校验设备存在于 `/dev/disk/by-id` 中

   ```
   > ls -l /dev/disk/by-id/
   ```
4. 通过输入以下命令，校验新设备是否出现在日志中

   ```
   > sudo journalctl -r
   ```
5. 使用文本编辑器，在 `/etc/multipath.conf` 文件中为该设备添加新的别名定义，例如 `data_vol3`。

   例如，如果 UUID 是 `36006016088d014006e98a7a94a85db11`，则进行以下更改：

   ```
   defaults {
   ```

```
    user_friendly_names   yes
  }
multipaths {
    multipath {
        wwid    36006016088d014006e98a7a94a85db11
        alias  data_vol3
        }
  }
```

6. 通过输入以下命令，为该设备创建分区表

```
> fdisk /dev/disk/by-id/dm-uuid-mpath-<UUID>
```

使用设备的 WWID（如 36006016088d014006e98a7a94a85db11）替换 UUID。

7. 通过输入以下命令触发 udev

```
> sudo echo 'add' > /sys/block/DM_DEVICE/uevent
```

例如，要为 dm-8 上的分区生成设备映射程序设备，请输入

```
> sudo echo 'add' > /sys/block/dm-8/uevent
```

8. 在设备 /dev/disk/by-id/dm-uuid-mpath-UUID_partN 上创建文件系统。根据您选择的文件系统，您可以使用以下其中一个命令实现此目的：**mkfs.btrfs** **mkfs.ext3**、**mkfs.ext4** 或 **mkfs.xfs**。有关细节，请参见相应的手册页。用实际的 UUID 和分区编号替换 UUID_partN，例如 36006016088d014006e98a7a94a85db11_part1。

9. 输入以下命令为新分区创建标签：

```
> sudo tune2fs -L LABELNAME /dev/disk/by-id/dm-uuid-UUID_partN
```

用实际的 UUID 和分区编号替换 UUID_partN，例如 36006016088d014006e98a7a94a85db11_part1。用您选择的标签替换 LABELNAME。

10. 输入以下命令以重配置 DM-MPIO，使其读取别名：

```
> sudo multipathd -k'reconfigure'
```

11. 通过输入以下命令，校验该设备已由 `multipathd` 识别

```
> sudo multipath -ll
```

12. 使用文本编辑器在 `/etc/fstab` 文件中添加新的挂载项。

    此时，您在上一步中创建的别名还不在 `/dev/disk/by-label` 目录中。为 `/dev/dm-9` 路径添加一个挂载项，然后在下次重引导到以下项目之前更改该项目

```
LABEL=LABELNAME
```

13. 创建要作为挂载点的目录，然后挂载设备。

18.14.3 查看多路径 I/O 状态

查询多路径 I/O 状态会输出多路径映射的当前状态。

`multipath -l` 选项可显示从上次运行路径检查程序以来的当前路径状态。它不会运行路径检查程序。

`multipath -ll` 选项运行路径检查程序，更新路径信息，然后显示当前状态信息。此命令始终显示关于路径状态的最新信息。

```
> sudo multipath -ll
3600601607cf30e00184589a37a31d911
[size=127 GB][features="0"][hwhandler="1 emc"]

\_ round-robin 0 [active][first]
  \_ 1:0:1:2 sdav 66:240  [ready ][active]
  \_ 0:0:1:2 sdr  65:16   [ready ][active]

\_ round-robin 0 [enabled]
  \_ 1:0:0:2 sdag 66:0    [ready ][active]
  \_ 0:0:0:2 sdc  8:32    [ready ][active]
```

对于每个设备，它显示该设备的 ID、大小、功能和硬件处理程序。

在设备发现时，通向该设备的路径会自动分组为多个优先级组。一次只有一个优先级组是活动的。对于活动/活动配置，所有路径均在同一组中。对于活动/不活动配置，不活动的路径会放在另外的优先级组中。

会针对每个组显示下列信息：

- 用于在组内平衡 I/O 的日程安排策略，例如循环
- 该组为活动、已禁用还是已启用
- 该组是否是第一个（最高优先级）组
- 该组内包含的路径

会针对每个路径显示下列信息：

- 物理地址，格式为 `HOST:BUS:TARGET:LUN`，例如 1:0:1:2
- 设备节点名，例如 `sda`
- 主要:次要编号
- 设备的状态

注意：在多路径设置中使用 `iostat`

在多路径环境中，**`iostat`** 命令可能会导致意外结果。默认情况下，**`iostat`** 会过滤掉所有没有 I/O 的块设备。要使 **`iostat`** 显示**所有**设备，请使用：

```
iostat -p ALL
```

18.14.4 在错误状况下管理 I/O

如果所有路径同时失败，则可能需要通过启用 queue_if_no_path 来配置多路径以使 I/O 排队。否则，I/O 将在通过所有路径后立即失败。在某些情形下，即驱动程序、HBA 或光纤遇到欺骗性错误时，应该将 DM-MPIO 配置为在这些错误导致丢失所有路径时将所有 I/O 排队，并且永不向上散播错误。

在群集中使用多路径设备时，可以选择禁用 queue_if_no_path。这会自动使路径失败，而不是使 I/O 排队，并升级 I/O 错误以引发群集资源的故障转移。

因为启用 queue_if_no_path 会导致 I/O 无限排队（除非有一个路径恢复），所以请确保 **multipathd** 正在运行且在您的环境中起作用。否则，可能需要在受影响的多路径设备上永远停止 I/O，直到重引导或手动返回到故障转移而不是排队。

测试该情形：

1. 打开一个终端控制台。
2. 输入以下命令激活设备 I/O 的排队而非故障转移

   ```
   > sudo dmsetup message DEVICE_ID 0 queue_if_no_path
   ```

   将 DEVICE_ID 替换为您设备的 ID。值 0 代表扇区，在不需要扇区信息时使用该值。
   例如，输入：

   ```
   > sudo dmsetup message 3600601607cf30e00184589a37a31d911 0 queue_if_no_path
   ```

3. 输入以下命令返回到设备 I/O 的故障转移

   ```
   > sudo dmsetup message DEVICE_ID 0 fail_if_no_path
   ```

   该命令会立即导致所有排队的 I/O 失败。
   将 DEVICE_ID 替换为您设备的 ID。例如，输入

   ```
   > sudo dmsetup message 3600601607cf30e00184589a37a31d911 0 fail_if_no_path
   ```

在所有路径失败的情形下设置将 I/O 排队：

1. 打开一个终端控制台。
2. 在文本编辑器中打开 /etc/multipath.conf。
3. 取消注释 defaults 部分及其结尾括号，然后如下所示添加 default_features 设置：

   ```
   defaults {
     default_features "1 queue_if_no_path"
   }
   ```

4. 修改 /etc/multipath.conf 文件后，您必须运行 **dracut** -f 在系统中重新创建 initrd，然后重引导以使更改生效。

5. 当您准备好返回到设备 I/O 的故障转移时，请输入

```
> sudo dmsetup message MAPNAME 0 fail_if_no_path
```

将 MAPNAME 替换为该设备映射的别名或设备 ID。值 0 代表扇区，在不需要扇区信息时使用该值。
此命令会立即导致所有排队的 I/O 失败，并且将该错误传播到调用的应用程序。

18.14.5 解决停止的 I/O

如果所有路径同时失败，并且 I/O 排队并停止，请执行以下操作：

1. 在终端控制台提示符下输入以下命令：

```
> sudo dmsetup message MAPNAME 0 fail_if_no_path
```

将 MAPNAME 替换为设备的正确设备 ID 或映射别名。值 0 代表扇区，在不需要扇区信息时使用该值。
此命令会立即导致所有排队的 I/O 失败，并且将该错误传播到调用的应用程序。

2. 输入以下命令重新激活排队：

```
> sudo dmsetup message MAPNAME 0 queue_if_no_path
```

18.14.6 配置 IBM Z 设备的默认设置

对包含多路径功能的 IBM Z 设备进行的测试表明，应将 dev_loss_tmo 参数设置为无穷 (2147483647)，将 fast_io_fail_tmo 参数设置为 5 秒。如果您使用的是 IBM Z 设备，请按以下方式修改 /etc/multipath.conf 文件来指定这些值：

```
defaults {
       dev_loss_tmo 2147483647
       fast_io_fail_tmo 5
}
```

`dev_loss_tmo` 参数设置将某个多路径链接标记为无效之前需等待的秒数。如果路径失败，则已失败路径上的任何当前 I/O 将失败。默认值根据所使用的设备驱动程序而异。要使用驱动程序的内部超时，请将值设置为零 (0)。也可以将其设置为“infinity”或 2147483647，如此会将其设置为最大值 2147483647 秒（68 年）。

`fast_io_fail_tmo` 参数设置检测到链接问题时将 I/O 确定为失败之前需等待的时间。到达驱动程序的 I/O 将失败。如果 I/O 排在拥堵的队列中，则未到 `dev_loss_tmo` 时间且队列未疏通之前 I/O 不会失败。

如果您修改了 `/etc/multipath.conf` 文件，所做的更改只有在您更新多路径映射或重启动 `multipathd` 守护程序 (**`systemctl restart multipathd`**) 后才会应用。

18.14.7 将多路径与 NetApp 设备配合使用

使用适用于 NetApp 设备的多路径时，建议在 `/etc/multipath.conf` 文件中进行如下设置：

- 为 NetApp 设备全局设置以下参数的默认值：

```
max_fds max
queue_without_daemon no
```

- 在硬件表中为 NetApp 设备的以下参数设置默认值：

```
dev_loss_tmo infinity
fast_io_fail_tmo 5
features "3 queue_if_no_path pg_init_retries 50"
```

18.14.8 将 --noflush 与多路径设备配合使用

在多路径设备上运行时，应始终使用 `--noflush`选项。

例如，在执行重新装载表的脚本中，应继续使用 `--noflush` 选项以确保任何未解决的 I/O 不会被刷新，因为您需要多路径拓扑信息。

```
load
resume --noflush
```

18.14.9 根设备为多路径时的 SAN 超时设置

所有路径都已失败并已从系统中去除时，多路径设备上包含根目录 (/) 的系统可能会停止，因为系统会收到储存子系统（例如光纤通道储存阵列）发出的 dev_loss_tmo 超时通知。

如果系统设备配置了多路径，且多路径 no_path_retry 设置处于活动状态，则应相应地修改储存子系统的 dev_loss_tmo 设置，以确保所有路径都失败时不会删除任何设备。强烈建议您将 dev_loss_tmo 值设置为等于或大于多路径的 no_path_retry 设置。

建议按如下方式设置储存子系统的 dev_los_tmo：

```
<dev_loss_tmo> = <no_path_retry> * <polling_interval>
```

以下定义适用于多路径值：

- no_path_retry 是将路径视为丢失并停止 IO 排队前多路径 I/O 的重试次数。
- polling_interval 是路径检查的间隔时间（以秒为单位）。

以上多路径值都应该从 /etc/multipath.conf 配置文件进行设置。有关信息，请参见第 18.6 节 “多路径配置”。

18.15 MPIO 查错

本节对 MPIO 的一些已知问题以及可行的解决方案进行了说明。

18.15.1 在多路径设备上安装 GRUB2

在具有 Btrfs 的传统 BIOS 系统上，**grub2-install** 可能会失败，并显示**权限被拒**。要解决此问题，请确保以读写 (rw) 模式挂载 /boot/grub2/SUBDIR/ 子卷。SUBDIR 可以是 x86_64-efi 或 i386-pc。

18.15.2 启用多路径时系统在引导过程中退出到紧急外壳

在引导过程中，系统退出到紧急外壳，并显示类似如下的消息：

```
[  OK  ] Listening on multipathd control socket.
         Starting Device-Mapper Multipath Device Controller...
[  OK  ] Listening on Device-mapper event daemon FIFOs.
         Starting Device-mapper event daemon...
         Expecting device dev-disk-by\x2duuid-34be48b2\x2dc21...32dd9.device...
         Expecting device dev-sda2.device...
[  OK  ] Listening on udev Kernel Socket.
[  OK  ] Listening on udev Control Socket.
         Starting udev Coldplug all Devices...
         Expecting device dev-disk-by\x2duuid-1172afe0\x2d63c...5d0a7.device...
         Expecting device dev-disk-by\x2duuid-c4a3d1de\x2d4dc...ef77d.device...
[  OK  ] Started Create list of required static device nodes ...current kernel.
         Starting Create static device nodes in /dev...
[  OK  ] Started Collect Read-Ahead Data.
[  OK  ] Started Device-mapper event daemon.
[  OK  ] Started udev Coldplug all Devices.
         Starting udev Wait for Complete Device Initialization...
[  OK  ] Started Replay Read-Ahead Data.
         Starting Load Kernel Modules...
         Starting Remount Root and Kernel File Systems...
[  OK  ] Started Create static devices
[*     ] (1 of 4) A start job is running for dev-disk-by\x2du...(7s / 1min 30s)
[*     ] (1 of 4) A start job is running for dev-disk-by\x2du...(7s / 1min 30s)

...

Timed out waiting for device dev-disk-by\x2duuid-c4a...cfef77d.device.
[DEPEND] Dependency failed for /opt.
[DEPEND] Dependency failed for Local File Systems.
[DEPEND] Dependency failed for Postfix Mail Transport Agent.
Welcome to emergency shell
Give root password for maintenance
(or press Control-D to continue):
```

在此阶段，您将在 initrd 环境中的临时 **`dracut`** 紧急外壳中工作。要永久保留以下所述的配置更改，您需要在已安装系统的环境中执行这些更改。

1. 确定系统根 (`/`) 文件系统。检查 `/proc/cmdline` 的内容并查找 `root=` 参数。

2. 校验是否已挂载根文件系统：

```
> sudo systemctl status sysroot.mount
```

提示

默认情况下，**dracut** 会将根文件系统挂载到 /sysroot 之下。

从现在开始，我们假设根文件系统已挂载到 /sysroot 下。

3. 在 /sysroot 下挂载系统所需的文件系统，使用 **chroot** 命令转到该目录，然后挂载所有文件系统。例如：

```
> sudo for x in proc sys dev run; do mount --bind /$x /sysroot/$x; done
> sudo chroot /sysroot /bin/bash
> sudo mount -a
```

有关更多详细信息，请参见 《管理指南》，第 48 章 “常见问题及其解决方案”，第 48.5.2.3 节 “访问已安装系统”。

4. 按照以下过程中的建议更改多路径或 dracut 配置。请记得重构建 initrd 以包含修改。

5. 通过输入 **exit** 命令退出 **chroot** 环境，然后退出紧急外壳并按 Ctrl – D 来重引导服务器。

过程 18.1： **紧急外壳：将文件系统加入黑名单**

如果根文件系统不在多路径上，却启用了多路径，则需要此修复。在此类设置中，多路径会尝试为未加入黑名单的所有设备设置其路径。由于具有根文件系统的设备已挂载，它对于多路径而言是无法访问的，因而会导致路径设置失败。您可以在 /etc/multipath.conf 中将根设备加入黑名单来正确设置多路径，从而修复此问题。

1. 在紧急外壳中运行 **multipath -v2** 并确定根文件系统的设备。产生的输出如下所示：

```
# multipath -v2
Dec 18 10:10:03 | 3600508b1001030343841423043300400: ignoring map
```

| 和 : 之间的字符串是加入黑名单所需的 WWID。

2. 打开 /etc/multipath.conf 并添加以下内容：

```
blacklist {
  wwid "WWID"
}
```

使用在上一步中获取的 ID 替换 WWID。有关更多信息，请参见第 18.8 节 “将非多路径设备加入黑名单”。

3. 使用以下命令重构建 initrd：

```
> dracut -f --add-multipath
```

过程 18.2： 紧急外壳：重构建 initrd

如果 initrd 与系统之间的多路径状态（已启用或已禁用）不相同，则需要执行此修复过程。要修复此问题，请重构建 initrd：

- 如果已在系统中**启用**多路径，请通过以下命令重构建支持多路径的 initrd：

```
> dracut --force --add multipath
```

如果已在系统中**禁用**多路径，请通过以下命令重构建支持多路径的 initrd：

```
> dracut --force -o multipath
```

过程 18.3： 紧急外壳：重构建 initrd

如果 initrd 不包含用于访问网络挂接储存的驱动程序，则需要此修复。例如，如果系统是在没有多路径的情况下安装的，或者当添加或更换相应硬件时，就可能需要执行此修复过程。

1. 将所需的驱动程序添加至文件 /etc/dracut.conf.d/01-dist.conf 中的变量 force_drivers。例如，如果系统包含 hpsa 驱动程序访问的 RAID 控制器，且多路径设备连接到由 qla23xx 驱动程序访问的 QLogic 控制器，该项将会如下所示：

```
force_drivers+="hpsa qla23xx"
```

2. 使用以下命令重构建 initrd：

```
> dracut -f --add-multipath
```

3. 为了防止在挂接网络储存失败时系统引导到紧急模式，建议将挂载选项 `_netdev` 添加到 `/etc/fstab` 中的相应项。

18.15.3 升级到多路径 0.4.9 或更高版本后单个设备的 PRIO 设置不起作用

从 0.4.9 版开始，多路径工具使用 `/etc/multipath.conf` 文件 `defaults{}` 或 `devices{}` 部分中的 `prio` 设置。当为 `multipaths{}` 部分中的单个 `multipath` 定义指定关键字 `prio` 时，它将被忽略而无反馈。

多路径工具 0.4.8 允许 `multipaths{)` 部分单个 `multipath` 定义中的 prio 设置覆盖 `defaults{}` 或 `devices{}` 部分中的 `prio` 设置。

18.15.4 升级到 multipath-tools-0.4.9 或更高版本后带有参数的 PRIO 设置不起作用

从 `multipath-tools-0.4.8` 升级到 `multipath-tools-0.4.9` 时，若优先级排序程序需要参数，则 `/etc/multipath.conf` 文件中的 `prio` 设置会损坏。在 multipath-tools-0.4.9 中，使用关键字 `prio` 指定优先级排序程序，使用关键字 `prio_args` 为需要参数的优先级排序程序指定参数。而在早前版本中，优先级排序程序及其参数均在同一个 `prio` 行上指定。

例如，在 multipath-tools-0.4.8 中，可使用以下一行命令指定优先级排序程序及其参数。

```
prio "weightedpath hbtl [1,3]:.:.+:.+ 260 [0,2]:.:.+:.+ 20"
```

升级到 multipath-tools-0.4.9 或更高版本后，该命令会导致错误。出现的消息类似如下所示：

```
<Month day hh:mm:ss> | Prioritizer 'weightedpath hbtl [1,3]:.:.+:.+ 260
[0,2]:.:.+:.+ 20' not found in /lib64/multipath
```

要解决此问题，请使用文本剪辑器修改 `/etc/multipath.conf` 文件中的 `prio` 行。创建两行，其中 `prio` 行指定优先级排序程序，而下方的 `prio_args` 行指定其参数：

```
prio "weightedpath"
prio_args "hbtl [1,3]:.:.+:.+ 260 [0,2]:.:.+:.+ 20"
```

运行 **`sudo systemctl restart multipathd`** 重启动 `multipathd` 守护程序，以使更改生效。

18.15.5 技术信息文档

有关 SUSE Linux Enterprise Server 上多路径 I/O 问题的查错信息，请参见 SUSE 知识库中的下列技术信息文档 (TID)：

- Using LVM on local and SAN attached devices（在本地设备和挂接 SAN 的设备上使用 LVM） (https://www.suse.com/support/kb/doc.php?id=3617600)
- Using LVM on Multipath (DM MPIO) Devices（在多路径 (DM MPIO) 设备上使用 LVM) (https://www.suse.com/support/kb/doc.php?id=7007498)
- HOWTO: Add, Resize and Remove LUN without restarting SLES（操作指南：在不重启动 SLES 的情况下添加、调整大小和去除 LUN） (https://www.suse.com/support/kb/doc.php?id=7009660)

19 通过 NFS 共享文件系统

网络文件系统 (**NFS**) 是允许访问服务器上的文件的协议，访问方式与访问本地文件相似。

SUSE Linux Enterprise Server 会安装 NFS v4.2，后者引入了对稀疏文件、文件预分配、服务器端克隆和复制、应用程序数据块 (ADB) 和用于强制性访问控制 (MAC) 的带标签 NFS（客户端和服务器上均需要 MAC）的支持。

19.1 概览

网络文件系统 (NFS) 是久经考验且受到广泛支持的标准化网络协议，它允许在单独的主机之间共享文件。

网络信息服务 (NIS) 可用于在网络中进行集中式用户管理。将 NFS 和 NIS 结合使用可通过文件和目录权限在网络中进行访问控制。NFS 与 NIS 一起使用时网络面向用户是透明的。

在默认配置中，NFS 完全信任网络，因此会信任连接到可信网络的任何计算机。在可通过物理方式访问 NFS 服务器所信任的任何网络的任何计算机上，任何具有管理员特权的用户都可以访问该服务器提供的所有文件。

一般而言，此安全性级别非常适于以下情形：所信任的网络是真正的专用网络，通常局限于单个计算机机柜或机房，并且无法进行未经授权的访问。将整个子网作为一个整体信任的其他情形限制较多，需要更精密的信任机制。为了满足这些情形的需要，NFS 使用 **Kerberos** 基础架构来支持各种安全性级别。Kerberos 需要 NFSv4（默认使用该协议）。有关详细信息，请参见《安全和强化指南》, 第 7 章 “使用 Kerberos 进行网络身份验证”。

下面是 YaST 模块中使用的术语。

导出

由 NFS 服务器**导出**的目录，客户端可将其集成到系统中。

NFS 客户端

NFS 客户端是通过网络文件系统协议使用来自 NFS 服务器的 NFS 服务的系统。TCP/IP 协议已集成到 Linux 内核中；无需再安装任何其他软件。

NFS 服务器

NFS 服务器向客户端提供 NFS 服务。运行中的服务器依赖于以下守护程序：`nfsd` (worker)、`idmapd`（用于 NFSv4 的 ID 到名称映射，仅在某些场景下需要）、`statd`（文件锁定）和 `mountd`（挂载请求）。

NFSv3

NFSv3 是版本 3 实施，支持客户端身份验证的“旧版”无状态 NFS。

NFSv4

NFSv4 是新的版本 4 实施，支持通过 Kerberos 进行安全用户身份验证。NFSv4 只需要一个端口，因此，它比 NFSv3 更适合用于防火墙后的环境。

协议指定为 https://datatracker.ietf.org/doc/html/rfc3530 ↗。

pNFS

并行 NFS，属于 NFSv4 的一种协议扩展。任何 pNFS 客户端都可以直接访问 NFS 服务器上的数据。

重要：需要 DNS 的原因

从理论上讲，所有导出都可以仅使用 IP 地址来完成。为避免超时，您需要一个有效的 DNS 系统。至少为了日志记录目的也应使用 DNS，因为 `mountd` 守护程序执行反向查找。

19.2 安装 NFS 服务器

默认不会安装 NFS 服务器。要使用 YaST 安装 NFS 服务器，请依次选择软件 › 软件管理、模式，然后启用服务器功能部分的文件服务器选项。单击接受安装所需软件包。

与 NIS 一样，NFS 也是一个客户端/服务器系统。但是，一台计算机可充当这两种角色：它可以通过网络提供文件系统（导出），也可以从其他主机挂载文件系统（导入）。

注意：在导出服务器上本地挂载 NFS 卷

SUSE Linux Enterprise Server 上不支持在导出服务器本地挂载 NFS 卷。

19.3 配置 NFS 服务器

可通过 YaST 配置 NFS 服务器，也可以手动配置它。NFS 还可与 Kerberos 结合来进行身份验证。

19.3.1 使用 YaST 导出文件系统

使用 YaST 将网络中的某台主机转换为 NFS 服务器，此服务器可将目录和文件导出到所有有权访问它的主机或导出到某个组的所有成员。因此，无需在每台主机本地安装应用程序，服务器也能提供应用程序。

要设置此类服务器，请继续执行以下步骤：

过程 19.1： 设置 NFS 服务器

1. 启动 YaST 并选择网络服务 › NFS 服务器；请参见图 19.1 “NFS 服务器配置工具”。系统会提示您安装其他软件。

图 19.1： NFS 服务器配置工具

2. 单击启动单选按钮。

3. 如果 `firewalld` 在系统上处于活动状态，请单独为 NFS 配置 firewalld（请参见《安全和强化指南》，第 24 章 “掩蔽和防火墙”，第 24.4 节 “`firewalld`”）。YaST 尚不完全支持 `firewalld`，因此请忽略“防火墙不可配置”消息并继续。

4. 选中是否启用 NFSv4。如果您停用 NFSv4，YaST 将只支持 NFSv3。有关启用 NFSv2 的信息，请参见注意：NFSv2。

   - 如果选择 NFSv4，另外还请输入相应的 NFSv4 域名。`idmapd` 守护程序会使用此参数。Kerberos 设置需要该守护程序，当客户端无法处理数字用户名时，也需要使用该守护程序。如果您不运行 `idmapd` 或无任何特殊要求，请将它保留为 `localdomain`（默认值）。有关 `idmapd` 守护程序的详细信息，请参见 `/etc/idmapd.conf`。

5. 如果您需要安全访问服务器，请单击启用 GSS 安全性。先决条件是您的域中安装了 Kerberos 并且服务器和客户端都已采用 Kerberos 系统。 单击下一步继续执行下一个配置对话框。

6. 单击对话框上半部分中的添加目录以导出您的目录。

7. 如果您尚未配置允许的主机，系统会自动弹出另一个对话框及相应的选项，供您输入客户端信息。输入主机通配符（通常您可以保留默认值不变）。
   可以为每个主机设置四类主机通配符：单主机（名称或 IP 地址）、网络组、通配符（如 `*` 表示所有计算机都能访问服务器）和 IP 网络。
   有关这些选项的更多信息，请参见 `exports` 手册页。

8. 单击完成以完成配置。

19.3.2 手动导出文件系统

NFS 导出服务的配置文件是 `/etc/exports` 和 `/etc/sysconfig/nfs`。如果 NFSv4 服务器配置包含经过 Kerberos 身份验证的 NFS，或者客户端不能使用数字用户名，则除了这些文件外，还需要 `/etc/idmapd.conf`。

要启动或重启动服务，请运行命令 **`systemctl restart nfsserver`**。此命令还会重启动 NFS 服务器所需的 RPC 端口映射程序。

为确保 NFS 服务器始终在引导时启动，请运行 **`sudo systemctl enable nfsserver`**。

注意：NFSv4

NFSv4 是 SUSE Linux Enterprise Server 上可用的最新版 NFS 协议。现在，通过 NFSv4 导出所用的配置目录与通过 NFSv3 导出所用的目录相同。

在 SUSE Linux Enterprise Server 11 上，必须在 `/etc/exports` 中指定绑定挂载。该设置仍然受支持，但现在已弃用。

`/etc/exports`

`/etc/exports` 文件包含项列表。每个条目表示共享的目录以及共享的方式。`/etc/exports` 中的条目通常包含：

```
/SHARED/DIRECTORY   HOST(OPTION_LIST)
```

例如：

```
/export/data   192.168.1.2(rw,sync)
```

在此，使用 IP 地址 `192.168.1.2` 标识允许的客户端。您还可以使用主机名、表示一组主机的通配符（`*.abc.com`、`*` 等）或网络组 (`@my-hosts`)。

有关所有选项及其含义的详细说明，请参见 `/etc/exports` 的手册页 (**`man exports`**)。

如果您在 NFS 服务器运行时修改了 `/etc/exports`，则需使用 **`sudo systemctl restart nfsserver`** 命令重启动 NFS 服务器，以使更改生效。

`/etc/sysconfig/nfs`

`/etc/sysconfig/nfs` 文件包含一些决定 NFSv4 服务器守护程序行为的参数。请务必将参数 `NFS4_SUPPORT` 设置为 `yes`（默认值）。`NFS4_SUPPORT` 决定 NFS 服务器是否支持 NFSv4 导出和客户端。

如果您在 NFS 服务器运行时修改了 `/etc/sysconfig/nfs`，则需使用 **`sudo systemctl restart nfsserver`** 命令重启动 NFS 服务器，以使更改生效。

提示：挂载选项

在 SUSE Linux Enterprise Server 11 上，必须在 `/etc/exports` 中指定 `--bind` 挂载。该设置仍然受支持，但现在已弃用。现在，通过 NFSv4 导出所用的配置目录与通过 NFSv3 导出所用的目录相同。

注意：NFSv2

如果 NFS 客户端仍依赖于 NFSv2，请在服务器的 `/etc/sysconfig/nfs` 中设置以下几项启用该协议：

```
NFSD_OPTIONS="-V2"
MOUNTD_OPTIONS="-V2"
```

重启动服务后，请使用以下命令检查版本 2 是否可用：

```
> cat /proc/fs/nfsd/versions
+2 +3 +4 +4.1 +4.2
```

`/etc/idmapd.conf`

仅当使用 Kerberos 身份验证或客户端不能使用数字用户名时，才需要 `idmapd` 守护程序。自 Linux 内核 2.6.39 起，Linux 客户端可以使用数字用户名。`idmapd` 守护程序会将发送到服务器的 NFSv4 请求进行名称到 ID 的映射，并答复客户端。

如果需要，`idmapd` 需在 NFSv4 服务器上运行。客户端上的名称到 ID 映射将由以下软件包提供的 **`nfsidmap`** 来执行： `nfs-client`.

对于可能使用 NFS 来共享文件系统的计算机，请确保以统一的方式在这些计算机间为用户指定用户名和 ID (UID)。这可以使用 NIS、LDAP 或域中的任何统一的域身份验证机制来实现。

在 `/etc/idmapd.conf` 文件中，必须为客户端和服务器设置相同的 `Domain` 参数。如果您不确定，请在服务器和客户端文件中将域保留为 `localdomain`。配置文件样本如下：

```
[General]
Verbosity = 0
Pipefs-Directory = /var/lib/nfs/rpc_pipefs
Domain = localdomain

[Mapping]
Nobody-User = nobody
Nobody-Group = nobody
```

要启动 `idmapd` 守护程序，请运行 **`systemctl start nfs-idmapd`**。如果您在守护程序运行时修改了 `/etc/idmapd.conf`，则需使用 **`systemctl start nfs-idmapd`** 命令重启动守护程序，以使更改生效。

有关更多信息，请参见 `idmapd` 和 `idmapd.conf` 的手册页（`man idmapd` 和 `man idmapd.conf`）。

19.3.3 采用 Kerberos 的 NFS

要对 NFS 使用 Kerberos 身份验证，必须启用通用安全服务 (GSS)。在初始 YaST NFS 服务器对话框中选择启用 GSS 安全。必须具有一个有效的 Kerberos 服务器才能使用此功能。YaST 不会设置服务器，而只使用所提供的功能。要使用 Kerberos 进行身份验证，除了 YaST 配置外，至少还须完成以下步骤才能运行 NFS 配置：

1. 请确保服务器和客户端都在同一 Kerberos 域中。它们必须访问相同的 KDC（密钥分发中心）服务器并共享其 `krb5.keytab` 文件（在任何计算机上的默认位置是 `/etc/krb5.keytab`）。有关 Kerberos 的更多信息，请参见《安全和强化指南》，第 7 章 “使用 Kerberos 进行网络身份验证”。
2. 在客户端上运行 **`systemctl start rpc-gssd.service`** 启动 gssd 服务。
3. 在服务器上运行 **`systemctl start rpc-svcgssd.service`** 启动 svcgssd 服务。

要进行 Kerberos 身份验证，也需要在服务器上运行 `idmapd` 守护程序。有关详细信息，请参见`/etc/idmapd.conf`。

有关配置采用 Kerberos 的 NFS 的更多信息，请参见第 19.6 节 “更多信息” 中的链接。

19.4 配置客户端

要将主机配置为 NFS 客户端，无需安装其他软件。将默认安装所有需要的软件包。

19.4.1 使用 YaST 导入文件系统

授权用户可以用 YaST NFS 客户端模块从 NFS 服务器将 NFS 目录挂载本地文件树。按如下所示继续：

过程 19.2： 导入 NFS 目录

1. 启动 YaST NFS 客户端模块。
2. 单击 NFS 共享选项卡中的添加。输入 NFS 服务器的主机名、要导入的目录以及要在本地的哪个挂载点挂载此目录。
3. 使用 NFSv4 时，在 NFS 设置选项卡中选择启用 NFSv4。另外，NFSv4 域名必须包含 NFSv4 服务器所用的相同值。默认域为 `localdomain`。
4. 要对 NFS 使用 Kerberos 身份验证，必须启用 GSS 安全性。选择启用 GSS 安全。
5. 如果您使用防火墙并希望允许从远程计算机访问服务，请启用 NFS 设置选项卡中的在防火墙上打开端口。防火墙状态将显示在复选框旁边。
6. 单击确定保存更改。

配置写入`/etc/fstab`，并将挂载指定的文件系统。当您稍后启动 YaST 配置客户端时，它还将读取此文件中的现有配置。

提示：NFS 用作根文件系统

在通过网络以 NFS 共享形式挂载根分区的（无磁盘）系统中，配置可供访问 NFS 共享的网络设备时需保持谨慎。

关闭或重引导系统时，默认的处理顺序是先关闭网络连接，然后卸载根分区。对于 NFS 根文件系统，此顺序会导致问题，因为在已停用与 NFS 共享的网络连接的情况下，根分区无法完全卸载。为防止系统停用相关的网络设备，请按《管理指南》，第 23 章 “基本网络知识”，第 23.4.1.2.5 节 “激活网络设备”中所述打开网络设备配置选项卡，然后在设备激活窗格中选择通过 NFSroot。

19.4.2 手动导入文件系统

手动从 NFS 服务器导入文件系统的先决条件是运行 RPC 端口映射器。`nfs` 服务负责正确启动该程序；因此，请以 `root` 身份输入 **`systemctl start nfs`** 来启动该服务。然后就可以像本地分区那样使用 **`mount`** 将远程文件系统挂载到文件系统中：

```
> sudo mount HOST:REMOTE-PATHLOCAL-PATH
```

例如，要从 `nfs.example.com` 计算机导入用户目录，请使用：

```
> sudo mount nfs.example.com:/home /home
```

要定义客户端到 NFS 服务器的 TCP 连接计数，可以使用 **`mount`** 命令的 `nconnect` 选项。您可以指定介于 1 和 16 之间的任何数字，其中 1 是默认值（如果未指定挂载选项）。

仅会在第一次挂载过程中对特定 NFS 服务器应用 `nconnect` 设置。如果同一客户端对同一 NFS 服务器执行挂载命令，则将共享所有已建立的连接，而不会建立新的连接。要更改 `nconnect` 设置，必须卸载到特定 NFS 服务器的**所有**客户端连接。然后，您可以为 `nconnect` 选项定义一个新值。

您可以在 **`mount`** 的输出或文件 `/proc/mounts` 中找到当前有效的 `nconnect` 值。如果挂载选项没有值，则在挂载期间不会使用该选项，而是使用默认值 **1**。

注意：连接数与 nconnect 定义的不同

由于您可以在第一次挂载后关闭和打开连接，因此实际连接计数不必与 `nconnect` 的值相同。

19.4.2.1 使用自动挂载服务

autofs 守护程序可用于自动挂载远程文件系统。请在 `/etc/auto.master` 文件中添加以下条目：

```
/nfsmounts /etc/auto.nfs
```

如果 `auto.nfs` 文件正确填充，`/nfsmounts` 目录将作为客户端上所有 NFS 挂载的 root 目录。选择 `auto.nfs` 这个名称是为了方便起见，您可以选择任何名称。在 `auto.nfs` 中为所有 NFS 挂载添加条目，如下所示：

```
localdata -fstype=nfs server1:/data
nfs4mount -fstype=nfs4 server2:/
```

以 root 身份运行 **systemctl start autofs** 来激活该设置。对于此示例，/nfsmounts/localdata，server1 的 /data 目录将通过 NFS 挂载，server2 的 /nfsmounts/nfs4mount 将通过 NFSv4 挂载。

如果在 autofs 服务正在运行时编辑了 /etc/auto.master 文件，则必须使用 **systemctl restart autofs** 重启动自动装载器才能使更改生效。

19.4.2.2 手动编辑 /etc/fstab

通常，/etc/fstab 中的 NFSv3 挂载项如下：

```
nfs.example.com:/data /local/path nfs rw,noauto 0 0
```

对于 NFSv4 挂载，请在第三列中使用 nfs4 而不是 nfs：

```
nfs.example.com:/data /local/pathv4 nfs4 rw,noauto 0 0
```

noauto 选项可禁止在启动时自动挂载文件系统。如果您要手动安装各文件系统，可以缩短只指定挂载点的安装命令：

```
> sudo mount /local/path
```

注意：启动时挂载

如果您没有输入 noauto 选项，系统的 init 脚本将在启动时处理这些文件系统的挂载。

19.4.3 并行 NFS (pNFS)

NFS 是最老的协议之一，开发于上世纪八十年代。因此，如果您要共享小文件，NFS 通常能够满足需求。但是，当您要传送大文件或有大量的客户端要访问数据时，NFS 服务器会成为瓶颈，严重影响系统性能。这是因为文件迅速变大，而以太网的相对速度没有完全跟上这一变化。

当您向普通 NFS 服务器请求文件时，服务器会查找文件元数据、收集所有数据并通过网络将数据传送到您的客户端。但是，无论文件的大小如何，性能瓶颈都会凸显出来：

- 如果是小文件，则大部分时间都花在收集元数据上。
- 如果是大文件，则大部分时间花在将数据从服务器传送到客户端上。

pNFS 或并行 NFS 则突破了此种限制，因为它将文件系统元数据从数据位置分离出来。因此，pNFS 需要两类服务器：

- 一个**元数据**或**控制服务器**，用于处理所有非数据通讯
- 一个或多个**储存服务器**，用于存放数据

元数据和储存服务器组成单独一个逻辑 NFS 服务器。当客户端要读取或写入时，元数据服务器会告诉 NFSv4 客户端使用哪个储存服务器访问文件块。客户端可以直接访问该服务器上的数据。

SUSE Linux Enterprise Server 仅在客户端上支持 pNFS。

19.4.3.1 使用 YaST 配置 pNFS 客户端

请执行过程 19.2 “导入 NFS 目录”中所述的步骤，但选中 pNFS (v4.2) 复选框以及可选的 NFSv4 共享。YaST 会执行所有必需的步骤，并且会在文件 `/etc/exports` 中写入所有必要选项。

19.4.3.2 手动配置 pNFS 客户端

请参阅第 19.4.2 节 “手动导入文件系统”着手配置。大多数配置通过 NFSv4 服务器完成。对于 pNFS，唯一的区别是将 `minorversion` 选项和元数据服务器 MDS_服务器添加到您的 **`mount`** 命令：

```
> sudo mount -t nfs4 -o minorversion=1 MDS_SERVER MOUNTPOINT
```

为方便调试，请更改 `/proc` 文件系统中的值：

```
> sudo echo 32767 > /proc/sys/sunrpc/nfsd_debug
```

```
> sudo echo 32767 > /proc/sys/sunrpc/nfs_debug
```

19.5 管理 NFSv4 访问控制列表

在 Linux 中，除了针对用户、组和其他人 (ugo) 的读取、写入、执行 (rwx) 这些简单标志之外，各访问控制列表 (ACL) 没有统一的标准。控制能力相对较好的一个选择是《Draft POSIX ACLs》（POSIX ACL 草稿），它尚未得到 POSIX 的正式标准化。另一个选择是 NFSv4 ACL，它是 NFSv4 网络文件系统的一部分，目的是为了在 Linux 上的 POSIX 系统与 Microsoft Windows 上的 WIN32 系统之间提供适当的兼容性。

NFSv4 ACL 不足以正确实施草稿 POSIX ACL，因此未进行在 NFSv4 客户端上映射 ACL 访问的尝试（比如使用 **setfacl**）。

使用 NFSv4 时，即使在仿真模式下也无法使用草稿 POSIX ACL，并且需要直接使用 NFSv4 ACL。这意味着，虽然 **setfacl** 能够在 NFSv3 上运行，却不能在 NFSv4 上运行。为了能够在 NFSv4 文件系统上使用 NFSv4 ACL，SUSE Linux Enterprise Server 提供了 `nfs4-acl-tools` 软件包，其中包含下列各项：

- **nfs4-getfacl**
- **nfs4-setfacl**
- **nfs4-editacl**

这些操作的运作方式与用于检查和修改 NFSv4 ACL 的 **getfacl** 和 **setfacl** 类似。仅当 NFS 服务器上的文件系统提供对 NFSv4 ACL 的全面支持时，这些命令才有效。虽然某些访问控制项 (ACE) 的特定组合可能在客户端中不可用，但客户端上运行的程序都将受到服务器施加的任何限制的影响。

不支持在输出 NFS 服务器上本地挂载 NFS 卷。

附加信息

有关信息，请参见 Introduction to NFSv4 ACLs（NFSv4 ACL 简介，网址为 http://wiki.linux-nfs.org/wiki/index.php/ACLs#Introduction_to_NFSv4_ACLs）。

19.6 更多信息

除了 **exports**、**nfs** 和 **mount** 的手册页外，还可在 /usr/share/doc/packages/nfsidmap/README 中找到关于配置 NFS 服务器和客户端的信息。有关更多联机文档，请参见以下网站：

- 有关网络安全的一般信息，请参见《安全和强化指南》，第 24 章 “掩蔽和防火墙”。
- 如果您需要自动挂载 NFS 导出，请参见第 21.4 节 “自动挂载 NFS 共享”。
- 有关使用 AutoYaST 配置 NFS 的更多细节，请参见《AutoYaST 指南》，第 4 章 “配置和安装选项”，第 4.20 节 “NFS 客户端和服务器”。
- 有关使用 Kerberos 保护 NFS 导出的说明，请参见《安全和强化指南》，第 7 章 “使用 Kerberos 进行网络身份验证”，第 7.6 节 “Kerberos 和 NFS”。
- 在 SourceForge (http://nfs.sourceforge.net/) 上联机查找详细的技术文档。

19.7 收集信息以供 NFS 查错

19.7.1 常见查错

某些情况下，您可以通过读取生成的错误消息并查看 /var/log/messages 文件来了解 NFS 中的问题。但很多时候，错误消息和 /var/log/messages 中提供的信息不够详细。在这些情况下，可通过再现问题时捕获网络数据包来充分了解大部分 NFS 问题。

明确定义问题。通过以各种方式测试系统并确定问题的发生时间来检查问题。隔离会导致问题的最简单步骤。然后尝试按照下面的过程再现问题。

过程 19.3： 再现问题

1. 捕获网络数据包。在 Linux 上，可以使用 tcpdump 软件包提供的 **tcpdump** 命令。**tcpdump** 语法的示例如下：

```
tcpdump -s0 -i eth0 -w /tmp/nfs-demo.cap host x.x.x.x
```

位置:

s0

防止数据包截断

eth0

应替换为将传递数据包的本地接口的名称。您可以使用 `any` 值同时捕获所有接口，但使用此属性通常会导致数据质量下降并造成分析混乱。

w

指定要写入的捕获文件的名称。

x.x.x.x

应替换为 NFS 连接另一端的 IP 地址。例如，在 NFS 客户端获取 **`tcpdump`** 时，请指定 NFS 服务器的 IP 地址，反之亦然。

注意

在某些情况下，只需在 NFS 客户端或 NFS 服务器任一端捕获数据就足够了。但如果不确定端到端网络的完整性，则通常需要在两端捕获数据。

请不要关闭 **`tcpdump`** 进程并继续下一步。

2. （可选）如果问题发生在 **`nfs mount`** 命令本身执行过程中，您可以尝试使用 **`nfs mount`** 命令的高详细程度选项 (`-vvv`) 来获得更多输出。

3. （可选）获取再现方法的 **`strace`**。再现步骤的 **`strace`** 精确记录了发生系统调用的确切时间。此信息可用于进一步确定您应关注 `tcpdump` 中的哪些事件。
例如，如果您发现在 NFS 挂载上执行 **mycommand --param** 命令失败，则可以使用以下命令来 **`strace`** 命令：

```
strace -ttf -s128 -o/tmp/nfs-strace.out mycommand --param
```

如果您未获得任何再现步骤的 **`strace`**，请注意问题的再现时间。检查 `/var/log/messages` 日志文件以隔离问题。

4. 一旦问题再现，按 CTRL – c 来停止在终端中运行的 **tcpdump**。如果 **strace** 命令导致挂起，还需终止 **strace** 命令。

5. 现在，具有分析数据包跟踪和 **strace** 数据经验的管理员可以检查 `/tmp/nfs-demo.cap` 和 `/tmp/nfs-strace.out` 中的数据。

19.7.2 高级 NFS 调试

重要：高级调试适用于专家

请注意，以下部分仅适用于了解 NFS 代码的高技能 NFS 管理员。因此，请执行第 19.7.1 节 “常见查错” 中所述的第一步，以帮助缩小问题范围，并告知专家要了解更深入的细节可能需要哪些方面的调试代码（如果有）。

可启用各种调试代码来收集额外的 NFS 相关信息。不过，调试消息非常晦涩难懂，并且数据量巨大，因此使用调试代码可能会影响系统性能。它甚至有可能对系统产生的影响大到足以防止问题的发生。大多数情况下都不需要调试代码输出，对于不太熟悉 NFS 代码的人来说，通常也没什么用。

19.7.2.1 使用 rpcdebug 激活调试

rpcdebug 工具可让您设置和清除 NFS 客户端和服务器调试标志。如果在 SLE 中无法访问 **rpcdebug** 工具，可以从 `nfs-client` 或 `nfs-kernel-server` 软件包为 NFS 服务器安装此工具。

要设置调试标志，请运行：

```
rpcdebug -m module -s flags
```

要清除调试标志，请运行：

```
rpcdebug -m module -c flags
```

其中，`module` 可以是：

nfsd

NFS 服务器代码的调试

nfs

NFS 客户端代码的调试

nlm

NFS 锁管理器调试（在 NFS 客户端或 NFS 服务器端）。仅适用于 NFS v2/v3。

rpc

远程过程调用模块调试（在 NFS 客户端或 NFS 服务器端）。

有关 **rpcdebug** 命令详细用法的信息，请参见手册页：

```
man 8 rpcdebug
```

19.7.2.2 针对 NFS 所依赖的其他代码激活调试

NFS 活动可能依赖于其他相关服务，例如 NFS 挂载守护程序—**rpc.mountd**。您可以在 /etc/sysconfig/nfs 中为相关服务设置选项。

例如，/etc/sysconfig/nfs 包含以下参数：

```
MOUNTD_OPTIONS=""
```

要启用调试模式，必须使用 -d 选项，后跟以下任何值：all、auth、call、general 或 parse。

例如，以下代码可启用所有形式的 **rpc.mountd** 日志记录：

```
MOUNTD_OPTIONS="-d all"
```

有关所有可用选项，请参见手册页：

```
man 8 rpc.mountd
```

更改 /etc/sysconfig/nfs 后，需要重启动服务：

```
systemctl restart nfsserver  # for nfs server related changes
```

```
systemctl restart nfs  # for nfs client related changes
```

20 Samba

使用 Samba 可以将 Unix 计算机配置为 macOS、Windows 和 OS/2 计算机的文件和打印服务器。Samba 已经发展成为一个功能完备且相当复杂的产品。使用 YaST 或手动编辑配置文件来配置 Samba。

重要：将禁用 SMB1

从 Samba 版本 4.17 开始，SMB1 协议已在 SLE 中禁用，且不再受支持。

20.1 术语

以下是 Samba 文档和 YaST 模块中使用的一些术语。

SMB 协议

Samba 使用基于 NetBIOS 服务的 SMB（服务器消息块）协议。Microsoft 发布该协议的目的是让来自其他制造商的软件可以与运行 Microsoft 操作系统的服务器建立连接。Samba 是在 TCP/IP 协议的基础上实施 SMB 协议的，也就是说，所有客户端上都必须安装并启用 TCP/IP。

提示：IBM Z：NetBIOS 支持

IBM Z 仅支持基于 TCP/IP 的 SMB。这些系统上不提供 NetBIOS 支持。

CIFS 协议

CIFS（Common Internet File System，通用互联网文件系统）协议是 SMB 协议的早期版本，也称为 SMB1。CIFS 定义 TCP/IP 上使用的标准远程文件系统访问协议，使用户组能够通过互联网协同工作并共享文档。

SMB1 已被 SMB2 取代，后者最初是作为 Microsoft Windows Vista™ 的一部分发布的。SMB2 后来又被 Microsoft Windows 8™ 和 Microsoft Windows Server 2012 中的 SMB3 取代。在最近的 Samba 版本中，出于安全原因默认已禁用 SMB1。

NetBIOS

NetBIOS 是专用于名称解析和在网络上的计算机之间进行通讯的软件接口 (API)。它使连接到网络的计算机能够为自己保留名称。之后便可以根据名称对这些计算机进行寻址。没有任何中心进程来检查这些名称。网络上的任何计算机均可以保留所需数量的名称，前提是这些名称尚未使用。可以在不同网络协议的基础上实现 NetBIOS。一种相对简单、不可路由的实现称为 NetBEUI（常常与 NetBIOS API 混淆）。NetBIOS 也可以在 Novell IPX/SPX 协议上运行。从版本 3.2 开始，Samba 支持在 IPv4 和 IPv6 上运行 NetBIOS。

通过 TCP/IP 发送的 NetBIOS 名称与 `/etc/hosts` 中使用的名称或 DNS 定义的名称没有相同之处。NetBIOS 使用它自己的、完全独立的命名约定。但为了方便管理或本机使用 DNS，建议您使用与 DNS 主机名对应的名称。Samba 默认采用这种方式。

Samba 服务器

Samba 服务器向客户端提供 SMB/CIFS 服务和 NetBIOS over IP 命名服务。对于 Linux，Samba 服务器有三个守护程序可用：`smbd` 用于 SMB/CIFS 服务，`nmbd` 用于命名服务，`winbind` 用于身份验证。

Samba 客户端

Samba 客户端是一种能够通过 SMB 协议从 Samba 服务器使用 Samba 服务的系统。常用操作系统（例如 Windows 和 macOS）都支持 SMB 协议。必须在所有计算机上安装 TCP/IP 协议。Samba 提供适用于多种不同类型 UNIX 的客户端。对于 Linux，有一个用于 SMB 的内核模块，它允许在 Linux 系统级别上集成 SMB 资源。不需要对 Samba 客户端运行任何守护程序。

共享

SMB 服务器通过**共享**为客户端提供资源。共享是指服务器上的目录（包括其子目录）和打印机。通过**共享名称**可导出和访问共享。可以将共享名称设置为任何名称 — 不一定是导出目录的名称。共享打印机也有相应的名称。客户端可以根据共享目录和打印机的名称来访问它们。

按惯例，以美元字符 (`$`) 结尾的共享名称会隐藏起来，也就是说，当使用 Windows 计算机浏览可用共享时，它们将不会显示。

DC

域控制器 (DC) 是处理域中帐户的服务器。为了进行数据复制，在单个域中可以使用多个域控制器。

20.2 安装 Samba 服务器

要安装 Samba 服务器，请启动 YaST 并选择软件 › 软件管理。选择视图 › 模式，然后选择文件服务器。确认已安装完成安装进程所需的软件包。

20.3 启动和停止 Samba

（引导时）可以自动启动或停止 Samba 服务器，或者手动执行这两个操作。启动和停止策略是第 20.4.1 节 “使用 YaST 配置 Samba 服务器”中所述的 YaST Samba 服务器配置的一部分。

在命令行中使用 **`systemctl stop smb nmb`** 可停止 Samba 所需的服务，使用 **`systemctl start nmb smb`** 可启动它们。`smb` 服务会根据需要处理 `winbind`。

提示：winbind

`winbind` 是一项独立服务，同样也是以单独的 `samba-winbind` 软件包提供。

20.4 配置 Samba 服务器

SUSE® Linux Enterprise Server 中的 Samba 服务器可通过两种不同方式进行配置：用 YaST 或手动方式。手工配置可提供更详细的信息，但没有 YaST GUI 方便。

20.4.1 使用 YaST 配置 Samba 服务器

要配置 Samba 服务器，请启动 YaST 并选择网络服务 › Samba 服务器。

20.4.1.1 初始 Samba 配置

第一次启动此模块时，系统会启动 Samba 安装对话框，提示您做一些涉及服务器管理的基本设置。配置结束时，系统会提示您输入 Samba 管理员口令（Samba Root 口令）。以后启动时，会显示 Samba 配置对话框。

Samba 安装对话框包括两个步骤和详细设置（可选）：

工作组名或域名

在工作组名或域名中选择一个现有名称或输入一个新的名称，然后单击下一步。

Samba 服务器类型

在下一步中，指定服务器是应该充当主域控制器 (PDC)、备用域服务器 (BDC) 还是不充当域控制器。按下一步继续。

如果不想再继续详细的服务器配置，请单击确定确认。然后在最后的弹出框中，设置 Samba root 口令。

稍后可以在 Samba 配置对话框的启动、共享、身份、可信域和 LDAP 设置选项卡中更改所有设置。

20.4.1.2 在服务器上启用最新版本的 SMB 协议

在运行最新版 SUSE Linux Enterprise Server 或其他最新 Linux 版本的客户端上，默认已禁用不安全的 SMB1/CIFS 协议。但是，现有的 Samba 实例可能配置为仅使用 SMB1/CIFS 版协议处理共享。要与此类客户端交互，需将 Samba 配置为至少使用 SMB 2.1 协议来为共享提供服务。

在某些设置中只能使用 SMB1，例如，当这些设置依赖于 SMB1/CIFS 的 Unix 扩展时。这些扩展尚未移植到更高的协议版本。如果您遇到这种情况，请考虑更改设置，或参见第 20.5.2 节 “在客户端上挂载 SMB1/CIFS 共享”。

要更改设置，请在配置文件 `/etc/samba/smb.conf` 中设置全局参数 `server max protocol = SMB2_10`。有关所有可能值的列表，请参见 **`man smb.conf`**。

20.4.1.3 高级 Samba 配置

第一次启动 Samba 服务器模块时，在执行第 20.4.1.1 节 “初始 Samba 配置”中所述的两个初始步骤后，Samba 配置对话框即会显示。使用它调整您的 Samba 服务器配置。

编辑配置之后，单击确定保存设置。

20.4.1.3.1 启动服务器

在启动选项卡中，配置 Samba 服务器的启动。若想在每次系统引导时启动服务，请选择引导时。要激活手动启动，请选择手动。有关启动 Samba 服务器的更多信息，请参见第 20.3 节“启动和停止 Samba”。

在此选项卡中，还可以打开防火墙中的端口。为此应选择打开防火墙中的端口。如果有多个网络接口，则请通过单击防火墙细节、选择接口并单击确定来为 Samba 服务选择网络接口。

20.4.1.3.2 共享

在共享选项卡中，确定要激活的 Samba 共享。存在一些预定义的共享，例如主页和打印机。使用切换状态可在活动和不活动之间进行切换。单击添加可添加新共享，单击删除可删除选中共享。

允许用户共享目录使允许的组中的组成员可以与其他用户共享他们拥有的目录。例如，`users` 用于本地范围，`DOMAIN\Users` 用于域范围。该用户必须还确保文件系统权限允许访问。最大共享数可限制可以创建的共享的总数。要允许访问用户共享而无需身份验证，请启用允许来宾访问。

20.4.1.3.3 身份

在身份选项卡中，确定与主机关联的域（基本设置）以及是否在网络中使用备用主机名（NetBIOS 主机名）。可以使用 Microsoft Windows Internet Name Service (WINS) 进行名称解析。在这种情况下，激活使用 WINS 进行主机名解析，并确定是否通过 DHCP 检索 WINS 服务器。要设置专家全局设置或设置用户身份验证源，例如 LDAP 而不是 TDB 数据库，请单击高级设置。

20.4.1.3.4 可信域

要使其他域的用户能够访问您的域，在可信域选项卡中进行适当的设置。要添加新域，请单击添加。要除去所选的域，请单击删除。

20.4.1.3.5 LDAP 设置

在选项卡 LDAP 设置中，您可以确定要用于身份验证的 LDAP 服务器。要测试到 LDAP 服务器的连接，请单击测试连接。要设置专家 LDAP 设置或使用默认值，请单击高级设置。

有关 LDAP 配置的更多信息，请参见《安全和强化指南》, 第 6 章 “使用 389 Directory Server 的 LDAP”。

20.4.2 手动配置服务器

如果要将 Samba 用作服务器，请安装 `samba`。Samba 的主配置文件是 `/etc/samba/smb.conf`。可以将此文件分为两个逻辑部分。`[global]` 部分包含中央和全局设置。以下默认部分包含各个文件和打印机共享：

- [homes]
- [profiles]
- [users]
- [groups]
- [printers]
- [print$]

通过此方法，您可以设置不同的共享选项，或在 `[global]` 部分设置全局共享选项，这使得配置文件更容易理解。

20.4.2.1 global 部分

应该修改 `[global]` 部分的以下参数来满足网络设置的要求，以使其他计算机能在 Windows 环境中通过 SMB 访问 Samba 服务器。

`workgroup = WORKGROUP`

此行将 Samba 服务器指派到工作组。请将 `WORKGROUP` 替换为您网络环境的相应工作组。您的 Samba 服务器将出现在其 DNS 名称下，除非此名称已指派给网络中的其他计算机。如果 DNS 名称不可用，请使用 `netbiosname=MYNAME` 设置服务器名称。有关此参数的更多细节，请参见 `smb.conf` 手册页。

`os level = 20`

此参数确定您的 Samba 服务器是否会尝试成为其工作组的 LMB（本地主浏览器）。为了避免现有 Windows 网络因 Samba 服务器配置不当而中断，应选择非常低的值，如 2。有关此主题的更多信息，可以在《Samba 3 Howto》（Samba 3 操作指南）的“Network Browsing”（网络浏览）一章中找到；有关《Samba 3 Howto》的更多信息，请参见第 20.9 节 “更多信息”。

如果网络中没有其他 SMB 服务器（如 Windows 2000 服务器），并且您希望 Samba 服务器保留一份本地环境中存在的所有系统的列表，请将 `os level` 设置为一个较高的值（例如 65）。然后便可以选择您的 Samba 服务器作为本地网络的 LMB。

在更改此设置时，应认真考虑这样做对现有 Windows 网络环境的影响。应该首先在一个孤立网络中或一天中的非重要时间测试这些更改。

`wins support` **和** `wins server`

为了将您的 Samba 服务器集成到具有活动 WINS 服务器的现有 Windows 网络中，应启用 `wins server` 选项并将其值设置为 WINS 服务器的 IP 地址。

如果将您的 Windows 计算机连接到单独的子网，同时又需要它们互相通讯，则需要设置一个 WINS 服务器。要将 Samba 服务器转变为这样的 WINS 服务器，请设置选项 `wins support = Yes`。确保网络中只有一个 Samba 服务器启用了此设置。切勿在您的 `smb.conf` 文件中同时启用选项 `wins server` 和 `wins support`。

20.4.2.2 共享

以下示例描述了如何使 CD-ROM 驱动器和用户目录 (`homes`) 对 SMB 客户端可用。

[cdrom]

为了避免意外地使 CD-ROM 驱动器变得可用，应使用注释标记（在本例中是分号）取消这些行。删除第一列中的分号，以便与 Samba 共享 CD-ROM 驱动器。

例 20.1： CD-ROM 共享

```
[cdrom]
       comment = Linux CD-ROM
       path = /media/cdrom
       locking = No
```

`[cdrom]` 和 `comment`

`[cdrom]` 部分项是网络上的所有 SMB 客户端均可看到的共享的名称。可以添加一个附加 `comment` 来进一步描述此共享。

`path = /media/cdrom`

`path` 导出目录 `/media/cdrom`。

通过严格限制的默认配置，可使这种共享仅对此系统上存在的用户可用。如果应使此共享对所有用户可用，请向配置中添加一行 `guest ok = yes`。此设置为网络上的所有用户提供读权限。建议您认真处理此参数。在 `[global]` 部分使用此参数时更应如此。

`[homes]`

`[home]` 共享在这里特别重要。如果用户具有 Linux 文件服务器的有效帐户和口令以及自己的主目录，则他们可以连接到此共享。

例 20.2： [HOMES] 共享

```
[homes]
        comment = Home Directories
        valid users = %S
        browseable = No
        read only = No
        inherit acls = Yes
```

[homes]

只要没有其他共享使用连接到 SMB 服务器的用户的共享名称，就会使用 `[homes]` 共享指令动态生成一个共享。生成的共享名称就是用户名。

`valid users = %S`

成功建立连接后，会使用具体的共享名称替换 `%S`。对于 `[homes]` 共享，此名称始终是用户名。这样就可以将用户的共享访问权仅限制于此用户。

`browseable = No`

此设置使共享在网络环境中不可见。

`read only = No`

默认情况下，Samba 通过 `read only = Yes` 参数来禁止对任何已导出共享的写访问。要使共享可写，请设置值 `read only = No`，它与 `writable = Yes` 是等效的。

`create mask = 0640`

那些不是基于 MS Windows NT 的系统不能理解 Unix 权限的概念，所以它们在创建文件时不能指派权限。参数 `create mask` 定义了为新创建文件指派的访问权限。这仅适用于可写共享。实际上，此设置表示所有者具有读写权限，所有者的主组成员具有读取权限。`valid users = %S` 会阻止读取访问，即使该组具有读取权限也是如此。要使该组能够进行读或写访问，应取消 `valid users = %S` 一行。

警告：不要与 Samba 共享 NFS 载具

与 Samba 共享 `NFS` 挂载可能导致数据丢失，并且不支持这样做。请直接在文件服务器上安装 Samba，或者考虑使用替代方式，例如 `iSCSI`。

20.4.2.3 安全性级别

要提高安全性，可以使用口令来保护每个共享访问。SMB 提供以下检查许可权限的方式：

用户级安全性 (`security = user`)

此变体将用户的概念引入了 SMB。每个用户都必须使用自己的口令在服务器上注册。注册后，服务器可以根据用户名来授予访问各个已导出共享的权限。

ADS 级安全性 (`security = ADS`)

在该模式中，Samba 将在 Active Directory 环境中充当域成员。要在该模式中工作，运行 Samba 的计算机需要安装并配置 Kerberos。必须使用 Samba 将该计算机加入到 ADS 领域。此步骤可通过使用 YaST Windows 域成员资格模块完成。

域级安全性 (`security = domain`)

仅当计算机已加入 Windows NT 域时，此模式才能正常工作。Samba 会尝试将用户名和口令传递给 Windows 主要或备用域控制器来验证该信息，这与 Windows Server 采用的方式相同。它期望将加密口令参数设置为 `yes`。

选择共享、用户或域级安全性适用于整个服务器。无法既为服务器配置的某些共享提供共享级安全性，同时又为其他共享提供用户级安全性。但是，您可以为系统上每个已配置的 IP 地址运行单独的 Samba 服务器。

有关此主题的更多信息，可以在《Samba 3 操作指南》中找到。对于一个系统上的多个服务器，应注意选项 `interfaces` 和 `bind interfaces only`。

20.5 配置客户端

客户端只能通过 TCP/IP 访问 Samba 服务器。NetBEUI 和通过 IPX 的 NetBIOS 不能与 Samba 共用。

20.5.1 使用 YaST 配置 Samba 客户端

配置 Samba 客户端来访问 Samba 或 Windows 服务器上的资源（文件或打印机）。在网络服务 › Windows 域成员资格对话框中输入 Windows 或 Active Directory 域或工作组。如果激活也使用 SMB 信息进行 Linux 身份验证，则用户身份验证将在 Samba、Windows 或 Kerberos 服务器上运行。

单击专家设置获取高级配置选项。例如，使用挂载服务器目录表启用自动挂载服务器用户主目录和身份验证。这样用户就能访问他们在 CIFS 上的主目录。有关详细信息，请参见 `pam_mount` 手册页。

完成所有设置后，请确认对话框以完成配置。

20.5.2 在客户端上挂载 SMB1/CIFS 共享

第一个 SMB 网络协议版本 SMB1 或 CIFS 是不安全的早期协议，其开发者 Microsoft 现已將其弃用。出于安全原因，SUSE Linux Enterprise Server 上的 **`mount`** 命令默认只会使用较新的协议版本（即 SMB 2.1、SMB 3.0 或 SMB 3.02）挂载 SMB 共享。

但是，此项更改只会影响通过 `/etc/fstab` 执行的 **`mount`** 命令和挂载操作。您仍然可以通过明确要求的方式来使用 SMB1。请使用以下参数：

- **`smbclient`** 工具。
- SUSE Linux Enterprise Server 随附的 Samba 服务器软件。

在以下设置中，由于只能使用 SMB1，此项默认设置会导致连接失败：

- 使用不支持较新 SMB 协议版本的 SMB 服务器的设置。自 Windows 7 和 Windows Server 2008 开始，Windows 已推出 SMB 2.1 支持。
- 依赖于 SMB1/CIFS 的 Unix 扩展的设置。这些扩展尚未移植到更高的协议版本。

重要：系统安全性降低

遵循以下指导可以解决安全问题。有关这些问题的详细信息，请参见 https://blogs.technet.microsoft.com/filecab/2016/09/16/stop-using-smb1/。

尽快升级服务器以使用更安全的 SMB 版本。

有关在 SUSE Linux Enterprise Server 上启用适当协议版本的信息，请参见第 20.4.1.2 节 “在服务器上启用最新版本的 SMB 协议”。

如果您需要在当前的 SUSE Linux Enterprise Server 内核中启用 SMB1 共享，请将选项 `vers=1.0` 添加到所用的 **`mount`** 命令行中：

```
# mount -t cifs //HOST/SHARE /MOUNT_POINT -o username=USER_ID,vers=1.0
```

或者，您也可以在安装的 SUSE Linux Enterprise Server 中全局启用 SMB1 共享。要实现此目的，请在 `/etc/samba/smb.conf` 中 `[global]` 部分下方添加以下代码：

```
        client min protocol = CORE
```

20.6 将 Samba 用作登录服务器

在商务设置中，组织通常希望只允许已在中心实例上注册的用户进行访问。在基于 Windows 的网络中，此任务由主域控制器 (PDC) 来处理。您可以使用配置为 PDC 的 Windows 服务器，但也可借助 Samba 服务器完成此任务。例 20.3 “Smb.conf 中的 global 部分”中显示了必须在 `smb.conf` 的 `[global]` 部分设置的项。

例 20.3： SMB.CONF 中的 GLOBAL 部分

```
[global]
    workgroup = WORKGROUP
    domain logons = Yes
    domain master = Yes
```

需要准备适合 Windows 加密格式的用户帐户和口令。使用命令 **`smbpasswd`** `-a name` 可完成此任务。使用以下命令为计算机创建 Windows 域概念要求的域帐户：

```
useradd hostname
smbpasswd -a -m hostname
```

使用 **`useradd`** 命令可添加一个美元符号。命令 **`smbpasswd`** 在使用参数 `-m` 时自动插入此符号。带注释的配置示例 (`/usr/share/doc/packages/samba/examples/smb.conf.SUSE`) 包含自动执行此任务的设置。

```
add machine script = /usr/sbin/useradd -g nogroup -c "NT Machine Account" \
-s /bin/false %m
```

要确保 Samba 能够正确执行此脚本，请选择具有必需的管理员权限的 Samba 用户，并将其添加到 `ntadmin` 组中。然后可以使用以下命令为属于此 Linux 组的所有用户指派 `Domain Admin` 状态：

```
net groupmap add ntgroup="Domain Admins" unixgroup=ntadmin
```

20.7 配置了 Active Directory 的网络中的 Samba 服务器

如果您同时运行 Linux 服务器和 Windows 服务器，则可以构建两个独立的身份验证系统和网络，或者将服务器连接到使用一个中央身份验证系统的网络。由于 Samba 可以与 Active Directory 域协作，因此您可以将 SUSE Linux Enterprise Server 服务器加入 Active Directory (AD) 域。

要加入到 AD 域，请执行以下操作：

1. 作为 `root` 登录并启动 YaST。
2. 启动网络服务 › Windows 域成员。
3. 在 Windows 域成员资格屏幕上的域或工作组字段中输入要加入的域。

图 20.1： 确定 WINDOWS 域成员资格

4. 选中同时使用 SMB 信息进行 Linux 身份验证，以在服务器上使用 SMB 源进行 Linux 身份验证。
5. 单击确定并在提示时确认域连接。
6. 在 AD 服务器上提供 Windows Administrator 的口令，并单击确定。

现在您的服务器已经设置了从 Active Directory 域控制器获取认证数据。

提示：身份映射

在有多个 Samba 服务器的环境中，将不会采用一致的方式创建 UID 和 GID。指派给用户的 UID 将取决于用户首次登录的顺序，而这会导致在服务器间产生 UID 冲突。要解决此问题，您需要使用身份映射。有关详细信息，请参见https://www.samba.org/samba/docs/man/Samba-HOWTO-Collection/idmapper.html。

20.8 高级主题

本节介绍用于管理 Samba 套件的客户端部分与服务器部分的高级方法。

20.8.1 使用 systemd 自动挂载 CIFS 文件系统

在启动时可以使用 systemd 来挂载 CIFS 共享。为此，请执行如下操作：

1. 创建挂载点：

   ```
   > mkdir -p PATH_SERVER_SHARED_FOLDER
   ```

   其中 PATH_SERVER_SHARED_FOLDER 是后续步骤中提到的 /cifs/shared。

2. 创建 systemd 单元文件，并从上一步骤中指定的路径生成文件名（其中的“/”需替换为“-”），例如：

   ```
   > sudo touch /etc/systemd/system/cifs-shared.mount
   ```

   该文件包含以下内容：

   ```
   [Unit]
   Description=CIFS share from The-Server

   [Mount]
   What=//The-Server/Shared-Folder
   ```

```
Where=/cifs/shared
Type=cifs
Options=rw,username=vagrant,password=admin

[Install]
WantedBy=multi-user.target
```

3. 启用服务：

```
> sudo systemctl enable cifs-shared.mount
```

4. 启动服务：

```
> sudo systemctl start cifs-shared.mount
```

要校验该服务是否正在运行，请运行以下命令：

```
> sudo systemctl status cifs-shared.mount
```

5. 要确认 CIFS 共享路径是否可用，请尝试运行以下命令：

```
>  cd /cifs/shared
> ls -l

total 0
-rwxrwxrwx. 1 root    root    0 Oct 24 22:31 hello-world-cifs.txt
drwxrwxrwx. 2 root    root    0 Oct 24 22:31 subfolder
-rw-r--r--. 1 vagrant vagrant 0 Oct 28 21:51 testfile.txt
```

20.8.2 Btrfs 上的透明文件压缩

Samba 允许客户端针对 Btrfs 文件系统中的共享远程操作文件与目录压缩标志。Windows 资源管理器可让用户通过文件 › 属性 › 高级对话框来标记要进行透明压缩的文件/目录：

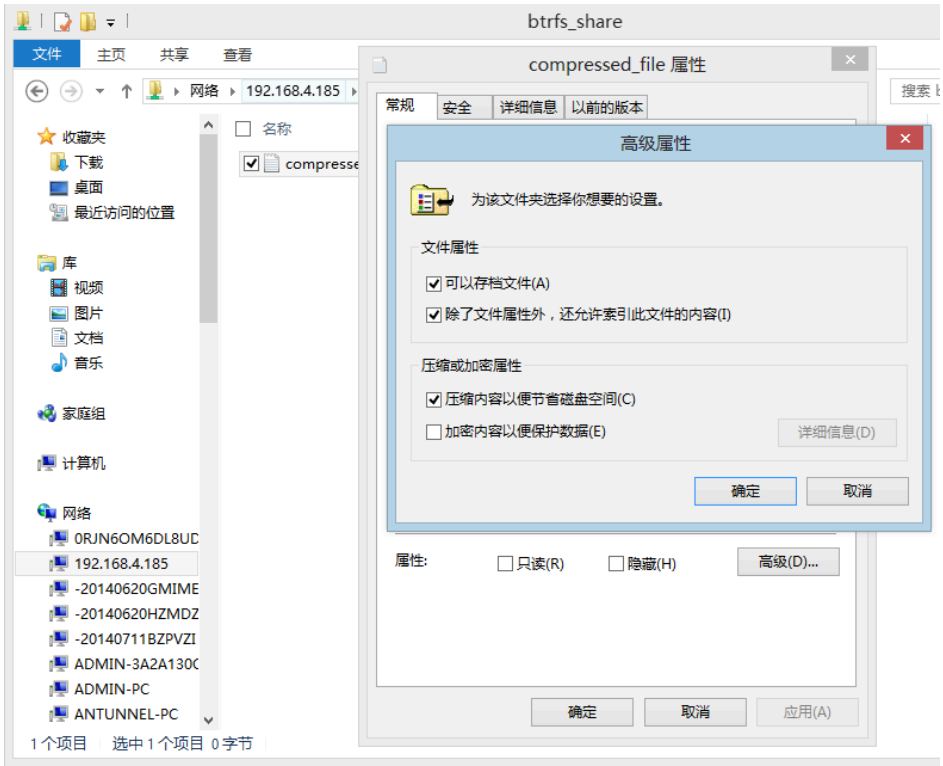


图 20.2： WINDOWS 资源管理器高级属性对话框

带有压缩标志的文件将以透明方式进行压缩，当用户访问或修改这些文件时，底层文件系统会将其解压缩。这通常可以节省储存容量，不过，在访问文件时会造成额外的 CPU 开销。除非新文件和目录是使用 FILE_NO_COMPRESSION 选项创建的，否则，它们将继承父目录的压缩标志。

Windows 资源管理器以不同的显示方式区分压缩文件和未压缩文件：

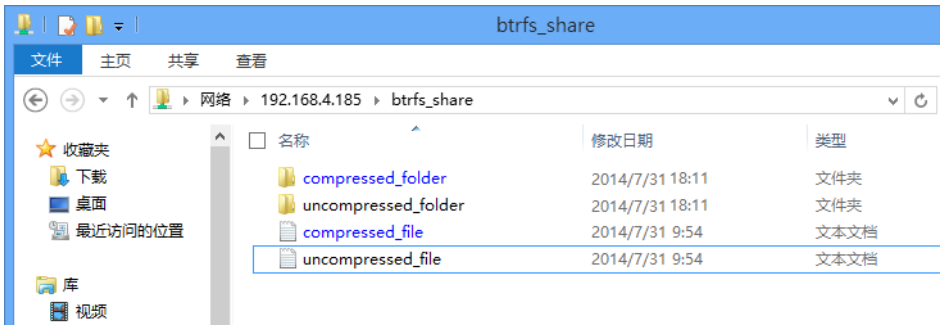


图 20.3： 列有压缩文件的 WINDOWS 资源管理器目录

要启用 Samba 共享压缩，您可以将以下内容

```
vfs objects = btrfs
```

手动添加到 `/etc/samba/smb.conf` 中的共享配置，或者使用 YaST：网络服务 › Samba 服务器 › 添加，然后选中使用 Btrfs 功能。

有关 Btrfs 上的压缩功能的一般概述，请参见第 1.2.2.1 节 “挂载压缩的 Btrfs 文件系统”。

20.8.3 快照

快照也称为阴影副本，是指某个文件系统子卷在特定时间点的状态副本。在 Linux 中，可以使用 Snapper 工具来管理这些快照。Btrfs 文件系统或精简配置的 LVM 卷支持快照。Samba 套件支持通过服务器端和客户端的 FSRVP 协议管理远程快照。

20.8.3.1 以前的版本

Samba 服务器上的快照可以作为先前版本的文件或目录向远程 Windows 客户端公开。

要在 Samba 服务器上启用快照，必须符合以下条件：

- SMB 网络共享位于 Btrfs 子卷上。
- SMB 网络共享路径中包含相关的 Snapper 配置文件。可以使用以下命令创建 snapper 文件

  ```
  > sudo snapper -c <cfg_name> create-config /path/to/share
  ```

  有关 Snapper 的详细信息，请参见《管理指南》, 第 10 章 “通过 Snapper 进行系统恢复和快照管理”。
- 必须允许相关用户访问快照目录树。有关更多信息，请参见 vfs_snapper 手册页 (**`man 8 vfs_snapper`**) 的 PERMISSIONS（权限）部分。

要支持远程快照，需要修改 `/etc/samba/smb.conf` 文件。要完成此操作，您可以选择 YaST › 网络服务 › Samba 服务器，或者使用以下命令手动增强相关的共享部分

```
vfs objects = snapper
```

请注意，只有在重启动 Samba 服务后，手动对 smb.conf 进行的更改才能生效：

```
> sudo systemctl restart nmb smb
```

图 20.4： 在启用快照的情况下添加新的 SAMBA 共享

经过配置后，可以通过 Windows 资源管理器中某个文件或目录的以前的版本选项卡访问 Snapper 为 Samba 共享路径创建的快照。

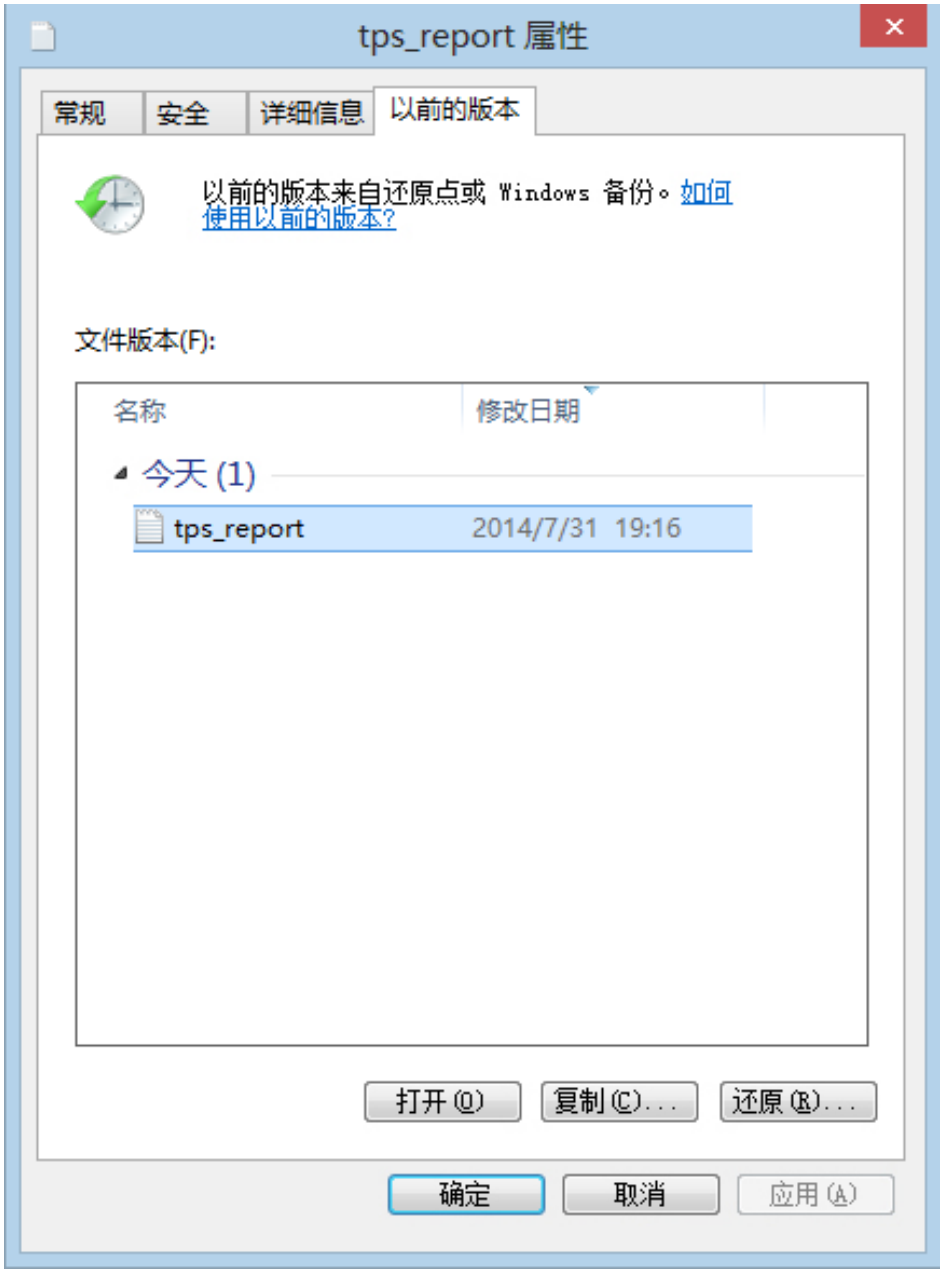


图 20.5： WINDOWS 资源管理器中的以前的版本选项卡

20.8.3.2 远程共享快照

默认情况下，只能在 Samba 服务器本地通过 Snapper 命令行实用程序或者使用 Snapper 的时间轴功能来创建和删除快照。

可将 Samba 配置为使用文件服务器远程 VSS 协议 (FSRVP) 处理远程主机发出的共享快照创建和删除请求。

除了第 20.8.3.1 节 “以前的版本”中所述的配置和先决条件以外，还需要在 `/etc/samba/smb.conf` 中指定以下全局配置：

```
[global]
rpc_daemon:fssd = fork
registry shares = yes
include = registry
```

然后，FSRVP 客户端（包括 Samba 的 **`rpcclient`** 以及 Windows Server 2012 **`DiskShadow.exe`**）便可以指示 Samba 为指定的共享创建或删除快照，并将该快照公开为新共享。

20.8.3.3 使用 `rpcclient` 从 Linux 中远程管理快照

`samba-client` 包中有一个 FSRVP 客户端，它可以远程请求 Windows/Samba 服务器创建并公开指定共享的快照。然后，您可以使用 SUSE Linux Enterprise Server 中的现有工具挂载公开的共享并备份其文件。向服务器发出的请求将使用 **`rpcclient`** 二进制文件发送。

例 20.4： 使用 rpcclient 请求 WINDOWS SERVER 2012 共享快照

以 `EXAMPLE` 域中管理员的身份连接到 `win-server.example.com` 服务器：

```
# rpcclient -U 'EXAMPLE\Administrator' ncacn_np:win-
server.example.com[ndr64,sign]
Enter EXAMPLE/Administrator's password:
```

检查 SMB 共享是否对于 **`rpcclient`** 可见：

```
# rpcclient $> netshareenum
netname: windows_server_2012_share
remark:
path:   C:\Shares\windows_server_2012_share
password:       (null)
```

检查 SMB 共享是否支持创建快照：

```
# rpcclient $> fss_is_path_sup windows_server_2012_share \
UNC \\WIN-SERVER\windows_server_2012_share\ supports shadow copy requests
```

请求创建共享快照：

```
# rpcclient $> fss_create_expose backup ro windows_server_2012_share
13fe880e-e232-493d-87e9-402f21019fb6: shadow-copy set created
13fe880e-e232-493d-87e9-402f21019fb6(1c26544e-8251-445f-be89-d1e0a3938777):
 \
\\WIN-SERVER\windows_server_2012_share\ shadow-copy added to set
13fe880e-e232-493d-87e9-402f21019fb6: prepare completed in 0 secs
13fe880e-e232-493d-87e9-402f21019fb6: commit completed in 1 secs
13fe880e-e232-493d-87e9-402f21019fb6(1c26544e-8251-445f-be89-d1e0a3938777):
 \
share windows_server_2012_share@{1C26544E-8251-445F-BE89-D1E0A3938777} \
exposed as a snapshot of \\WIN-SERVER\windows_server_2012_share\
```

确认服务器是否已公开快照共享：

```
# rpcclient $> netshareenum
netname: windows_server_2012_share
remark:
path:   C:\Shares\windows_server_2012_share
password:       (null)

netname: windows_server_2012_share@{1C26544E-8251-445F-BE89-D1E0A3938777}
remark: (null)
path:   \\?\GLOBALROOT\Device\HarddiskVolumeShadowCopy{F6E6507E-
F537-11E3-9404-B8AC6F927453}\Shares\windows_server_2012_share\
password:       (null)
```

尝试删除快照共享：

```
# rpcclient $> fss_delete windows_server_2012_share \
13fe880e-e232-493d-87e9-402f21019fb6 1c26544e-8251-445f-be89-d1e0a3938777
13fe880e-e232-493d-87e9-402f21019fb6(1c26544e-8251-445f-be89-d1e0a3938777):
 \
\\WIN-SERVER\windows_server_2012_share\ shadow-copy deleted
```

确认服务器是否已去除快照共享：

```
# rpcclient $> netshareenum
```

```
netname: windows_server_2012_share
remark:
path:   C:\Shares\windows_server_2012_share
password:       (null)
```

20.8.3.4 使用 DiskShadow.exe 从 Windows 中远程管理快照

您也可以从 Windows 客户端中管理 Linux Samba 服务器上 SMB 共享的快照。Windows Server 2012 提供了 **DiskShadow.exe** 实用程序，该实用程序可以使用与第 20.8.3.3 节 “使用 **rpcclient** 从 Linux 中远程管理快照” 中所述的 **rpcclient** 命令类似的方式管理远程共享。请注意，首先您需要妥善设置 Samba 服务器。

以下示例步骤描述了如何设置 Samba 服务器，以使 Windows 客户端能够管理其共享的快照。请注意，EXAMPLE 是测试环境中使用的 Active Directory 域，`fsrvp-server.example.com` 是 Samba 服务器的主机名，`/srv/smb` 是 SMB 共享的路径。

过程 20.1： SAMBA 服务器配置详细说明

1. 通过 YaST 加入到 Active Directory 域。有关详细信息，请参见第 20.7 节 “配置了 Active Directory 的网络中的 Samba 服务器” 。
2. 确保 Active Directory 域的 DNS 项正确无误：

   ```
   fsrvp-server:~ # net -U 'Administrator' ads dns register \
   fsrvp-server.example.com <IP address>
   Successfully registered hostname with DNS
   ```

3. 在 `/srv/smb` 位置创建 Btrfs 子卷

   ```
   fsrvp-server:~ # btrfs subvolume create /srv/smb
   ```

4. 为路径 `/srv/smb` 创建 Snapper 配置文件：

   ```
   fsrvp-server:~ # snapper -c <snapper_config> create-config /srv/smb
   ```

5. 创建路径为 `/srv/smb` 的新共享，并选中 YaST 的公开快照复选框。确保将以下代码段添加到 `/etc/samba/smb.conf` 的 global 部分，如第 20.8.3.2 节 “远程共享快照” 中所述：

```
[global]
 rpc_daemon:fssd = fork
 registry shares = yes
 include = registry
```

6. 使用 **`systemctl restart nmb smb`** 重启动 Samba

7. 配置 Snapper 权限：

```
fsrvp-server:~ # snapper -c <snapper_config> set-config \
ALLOW_USERS="EXAMPLE\\\\Administrator EXAMPLE\\\\win-client$"
```

确保还允许所有 `ALLOW_USERS` 实例访问 `.snapshots` 子目录。

```
fsrvp-server:~ # snapper -c <snapper_config> set-config SYNC_ACL=yes
```

重要：路径转义

请小心使用“\”转义！请转义两次，以确保 `/etc/snapper/configs/<snapper_config>` 中储存的值转义一次。

"EXAMPLE\win-client$" 对应于 Windows 客户端计算机帐户。对此帐户进行验证后，Windows 将发出初始 FSRVP 请求。

8. 授予 Windows 客户端帐户必要的特权：

```
fsrvp-server:~ # net -U 'Administrator' rpc rights grant \
"EXAMPLE\\win-client$" SeBackupPrivilege
Successfully granted rights.
```

不需要对 "EXAMPLE\Administrator" 用户执行上一条命令，因为已授予该用户特权。

过程 20.2： **WINDOWS 客户端设置和 `DiskShadow.exe` 的实际运用**

1. 引导 Windows Server 2012（示例主机名为 WIN-CLIENT）。

2. 就像在 SUSE Linux Enterprise Server 上那样，加入到同一个 Active Directory 域 EXAMPLE。

3. 重引导.

4. 打开 Powershell。

5. 启动 **DiskShadow.exe**，然后开始执行备份过程：

```
PS C:\Users\Administrator.EXAMPLE> diskshadow.exe
Microsoft DiskShadow version 1.0
Copyright (C) 2012 Microsoft Corporation
On computer:  WIN-CLIENT,  6/17/2014 3:53:54 PM

DISKSHADOW> begin backup
```

6. 指定阴影副本在程序退出、重设置和重引导期间持续存在。

```
DISKSHADOW> set context PERSISTENT
```

7. 检查指定的共享是否支持快照，然后创建一个快照：

```
DISKSHADOW> add volume \\fsrvp-server\sles_snapper

DISKSHADOW> create
Alias VSS_SHADOW_1 for shadow ID {de4ddca4-4978-4805-8776-cdf82d190a4a} set
 as \
 environment variable.
Alias VSS_SHADOW_SET for shadow set ID {c58e1452-c554-400e-a266-
d11d5c837cb1} \
 set as environment variable.

Querying all shadow copies with the shadow copy set ID \
 {c58e1452-c554-400e-a266-d11d5c837cb1}

 * Shadow copy ID = {de4ddca4-4978-4805-8776-cdf82d190a4a}
 %VSS_SHADOW_1%
     - Shadow copy set: {c58e1452-c554-400e-a266-d11d5c837cb1}
 %VSS_SHADOW_SET%
     - Original count of shadow copies = 1
     - Original volume name: \\FSRVP-SERVER\SLES_SNAPPER\ \
       [volume not on this machine]
     - Creation time: 6/17/2014 3:54:43 PM
```

```
    - Shadow copy device name:
      \\FSRVP-SERVER\SLES_SNAPPER@{31afd84a-44a7-41be-b9b0-751898756faa}
    - Originating machine: FSRVP-SERVER
    - Service machine: win-client.example.com
    - Not exposed
    - Provider ID: {89300202-3cec-4981-9171-19f59559e0f2}
    - Attributes:  No_Auto_Release Persistent FileShare

Number of shadow copies listed: 1
```

8. 完成备份过程：

```
DISKSHADOW> end backup
```

9. 创建快照后，尝试将它删除，并确认删除结果：

```
DISKSHADOW> delete shadows volume \\FSRVP-SERVER\SLES_SNAPPER\
Deleting shadow copy {de4ddca4-4978-4805-8776-cdf82d190a4a} on volume \
 \\FSRVP-SERVER\SLES_SNAPPER\ from provider \
{89300202-3cec-4981-9171-19f59559e0f2} [Attributes: 0x04000009]...

Number of shadow copies deleted: 1

DISKSHADOW> list shadows all

Querying all shadow copies on the computer ...
No shadow copies found in system.
```

20.9 更多信息

- **手册页：** 要查看随 `samba` 软件包一起安装的所有**手册**页的列表，请运行 **`apropos samba`**。使用 **`man`** 手册页名称打开任一手册页。
- **SUSE 特定的 README 文件：** 软件包 `samba-client` 包含文件 `/usr/share/doc/packages/samba/README.SUSE`。
- **其他软件包文档：** 使用 **`zypper install samba-doc`** 安装 `samba-doc` 软件包。

此文档将安装到 `/usr/share/doc/packages/samba` 中。其中包含手册页的 HTML 版本以及示例配置的库（例如 `smb.conf.SUSE`）。

- **联机文档:** https://wiki.samba.org/index.php/User_Documentation 上的 Samba Wiki 包含丰富的用户文档。

21 使用 Autofs 按需挂载

autofs 是一个可根据需要自动挂载指定目录的程序。它基于一个内核模块运行以实现高效率，并且可以同时管理本地目录和网络共享。这些自动挂载点仅会在被访问时挂载，一定时间内不活动后即会被卸载。这种按需行为可节省带宽，并实现比 /etc/fstab 管理的静态挂载更高的性能。虽然 autofs 是控制脚本，但 **automount** 才是实际执行自动挂载的命令（守护程序）。

21.1 安装

SUSE Linux Enterprise Server 上默认未安装 autofs。要使用它的自动装载功能，请先使用下面的命令安装该程序

```
> sudo zypper install autofs
```

21.2 配置

您需要使用 **vim** 等文本编辑器编辑 autofs 的配置文件来手动配置它。配置 autofs 有两个基本步骤 — **master** 映射文件和特定映射文件。

21.2.1 Master 映射文件

autofs 的默认 master 配置文件是 /etc/auto.master。可通过在 /etc/sysconfig/autofs 文件中更改 DEFAULT_MASTER_MAP_NAME 选项的值来更改其位置。以下是 SUSE Linux Enterprise Server 中默认 master 映射文件的内容：

```
#
# Sample auto.master file
# This is an automounter map and it has the following format
# key [ -mount-options-separated-by-comma ] location
```

```
# For details of the format look at autofs(5).❶
#
#/misc  /etc/auto.misc❷
#/net -hosts
#
# Include /etc/auto.master.d/*.autofs❸
#
#+dir:/etc/auto.master.d
#
# Include central master map if it can be found using
# nsswitch sources.
#
# Note that if there are entries for /net or /misc (as
# above) in the included master map any keys that are the
# same will not be seen as the first read key seen takes
# precedence.
#
+auto.master❹
```

❶ `autofs` 手册页 (**`man 5 autofs`**) 提供了许多有关该自动挂载器映射格式的重要信息。

❷ 虽然这些内容默认会被注释掉 (#)，但它依然是简单的自动挂载器映射语法示例。

❸ 如果您需要将 master 映射分割成几个文件，请将该行取消注释，并将映射（后缀为 `.autofs`）置于 `/etc/auto.master.d/` 目录中。

❹ `+auto.master` 可确保使用 NIS（请参见《安全和强化指南》, 第 4 章 “使用 NIS”，第 4.1 节 “配置 NIS 服务器” 了解 NIS 的更多信息）的用户仍可找到其 master 映射。

`auto.master` 中的项有三个字段，语法如下：

```
mount point      map name      options
```

mount point

要在其中挂载 `autofs` 文件系统的基本位置，例如 `/home`。

map name

挂载时所用映射源的名称。有关映射文件的语法，请参见第 21.2.2 节 “映射文件”。

options

这些选项（如指定）将作为默认值应用于给定映射中的所有项。

提示：更多信息

有关选用 `map-type`、`format` 和 `options` 的特定值的更多详细信息，请参见 auto.master 手册页 (**`man 5 auto.master`**)。

`auto.master` 中的下面一项指示 `autofs` 查看 `/etc/auto.smb`，并在 `/smb` 目录中创建挂载点：

```
/smb    /etc/auto.smb
```

21.2.1.1 直接挂载

直接挂载会在相关映射文件内的指定路径创建挂载点。这种方式不是在 `auto.master` 中指定挂载点，而是用 `/-` 替换挂载点字段。例如，下行指示 `autofs` 在 `auto.smb` 中指定的位置创建挂载点：

```
/-        /etc/auto.smb
```

提示：不含完整路径的映射

如果指定映射文件时未包含其完整本地或网络路径，系统会使用名称服务转换 (NSS) 配置寻找该映射文件。

```
/-        auto.smb
```

21.2.2 映射文件

重要：其他映射类型

虽然**文件**是使用 `autofs` 自动挂载的最常见的映射类型，但是还有其他一些类型。映射规范可以是命令的输出，也可以是 LDAP 或数据库中查询的结果。有关映射类型的更多详细信息，请参见 **`man 5 auto.master`** 手册页。

映射文件指定（本地或网络）来源位置，以及在本地挂载来源的挂载点。映射的一般格式与 master 映射相似。区别在于 **options** 位于 mount point 与 location 之间，而不是该项的末尾：

```
mount point      options      location
```

确保映射文件未标记为可执行文件。可通过执行 **chmod -x** MAP_FILE 去除可执行文件位。

mount point

指定将来源位置挂载到何处。这可以是要添加到 **auto.master** 中所指定基础挂载点的单个目录名称（所谓的间接挂载），也可以是挂载点的完整路径（直接挂载，请参见第 21.2.1.1 节 “直接挂载”）。

options

为相关项指定可选的挂载选项列表，挂载选项以逗号分隔。如果 `auto.master` 还包含此映射文件的选项，这些选项会附加在后面。

location

指定要挂载的文件系统来自何处。通常是 NFS 或 SMB 卷，常用表示法是主机名:路径名称。如果要挂载的文件系统以“/”开头（例如本地 `/dev` 项或 smbfs 共享），需要在前面加一个冒号“:”，例如 `:/dev/sda1`。

21.3 操作和调试

本节介绍如何控制 `autofs` 服务操作，以及如何在调整该自动挂载器操作时查看更多调试信息。

21.3.1 控制 autofs 服务

`autofs` 服务的操作由 `systemd` 控制。`autofs` 的 **systemctl** 命令的一般语法为

```
> sudo systemctl SUB_COMMAND autofs
```

其中，SUB_COMMAND 是下列项目之一：

enable

在引导时启动该自动挂载器守护程序。

start

启动该自动挂载器守护程序。

stop

停止该自动挂载器守护程序。自动挂载点将不再可访问。

status

打印 `autofs` 服务的当前状态以及相关日志文件的部分内容。

restart

停止然后启动该自动挂载器，以便终止所有正在运行的守护程序，然后再启动新的守护程序。

reload

检查当前的 `auto.master` 映射，重启动项已更改的守护程序，并为新项启动新守护程序。

21.3.2 调试自动挂载器问题

如果您在使用 `autofs` 挂载目录时遇到问题，手动运行 **`automount`** 守护程序并查看其输出消息将非常有用：

1. 停止 `autofs`。

```
> sudo systemctl stop autofs
```

2. 从一个终端的前台手动运行 **`automount`**，生成详细输出。

```
> sudo automount -f -v
```

3. 从另一个终端上尝试通过访问挂载点（例如，通过 **`cd`** 或 **`ls`**）挂载自动挂载文件系统。

4. 从第一个终端检查 **`automount`** 的输出，以了解有关挂载为何失败或者甚至为何未尝试挂载的更多信息。

21.4 自动挂载 NFS 共享

下面的过程说明了如何配置 `autofs` 以自动挂载网络上可用的 NFS 共享。该过程要用到前面提到的信息，并假设您熟悉 NFS 导出。有关 NFS 的更多信息，请参见第 19 章 “通过 NFS 共享文件系统”。

1. 编辑 master 映射文件 `/etc/auto.master`：

   ```
   > sudo vim /etc/auto.master
   ```

   在 `/etc/auto.master` 末尾为新的 NFS 挂载添加一条新项：

   ```
   /nfs      /etc/auto.nfs      --timeout=10
   ```

   此指令指示 `autofs` 基本挂载点是 `/nfs`，NFS 共享在 `/etc/auto.nfs` 映射中指定，并且此映射中的所有共享在 10 秒不活动后将自动卸载。

2. 为 NFS 共享创建新的映射文件：

   ```
   > sudo vim /etc/auto.nfs
   ```

   对每个 NFS 共享，`/etc/auto.nfs` 通常都会包含单独的一行。有关其格式，请参见第 21.2.2 节 “映射文件”。添加下行，指出挂载点及 NFS 共享网络地址：

   ```
   export      jupiter.com:/home/geeko/doc/export
   ```

   上面的行表示当收到请求时，系统会将 `jupiter.com` 主机上的 `/home/geeko/doc/export` 目录自动挂载到本地主机上的 `/nfs/export` 目录（`/nfs` 取自 `auto.master` 映射）。`/nfs/export` 目录将由 `autofs` 自动创建。

3. （选择性）如果您先前以静态方式挂载了该 NFS 共享，请将 `/etc/fstab` 中的相关行注释掉。该行应类似于：

   ```
   #jupiter.com:/home/geeko/doc/export /nfs/export nfs defaults 0 0
   ```

4. 重新装载 `autofs` 并检查它是否正常工作：

   ```
   > sudo systemctl restart autofs
   ```

```
# ls -l /nfs/export
total 20
drwxr-xr-x  5 1001 users 4096 Jan 14  2017 .images/
drwxr-xr-x 10 1001 users 4096 Aug 16  2017 .profiled/
drwxr-xr-x  3 1001 users 4096 Aug 30  2017 .tmp/
drwxr-xr-x  4 1001 users 4096 Apr 25 08:56 manual/
```

如果您能看到远程共享上的文件列表，则表示 `autofs` 工作正常。

21.5 高级主题

本节讨论的主题超出了 `autofs` 基本介绍的范畴 — 自动挂载网络上可用的 NFS 共享、在映射文件中使用通配符，以及特定于 CIFS 文件系统的信息。

21.5.1 /net 挂载点

如果您使用了许多 NFS 共享，这个助手挂载点将非常有用。`/net` 会根据需要自动挂载本地网络上的所有 NFS 共享。该项在 `auto.master` 文件中已经存在，因此，您只需将其取消注释，然后重启动 `autofs` 即可：

```
/net      -hosts
```

```
> sudo systemctl restart autofs
```

例如，如果您有名为 `jupiter` 的服务器以及名为 `/export` 的 NFS 共享，您可以在命令行上键入

```
> sudo cd /net/jupiter/export
```

来挂载它。

21.5.2 使用通配符自动挂载子目录

如果您的某个目录含有多个子目录，并且您需要单个自动挂载这些子目录（一般情况下，该目录是包含各个用户主目录的 `/home` 目录），`autofs` 提供了便捷的解决方案。

如果这些子目录是主目录，则在 `auto.master` 中添加下行：

```
/home      /etc/auto.home
```

现在，您需要在 `/etc/auto.home` 文件中添加正确的映射，以便自动挂载用户的主目录。一种方法是为每个目录创建单独的项：

```
wilber      jupiter.com:/home/wilber
penguin      jupiter.com:/home/penguin
tux      jupiter.com:/home/tux
[...]
```

这种方法非常麻烦，因为您需要在 `auto.home` 中管理用户列表。您可以使用星号“*”取代挂载点，使用符号“&”取代要挂载的目录。

```
*      jupiter:/home/&
```

21.5.3 自动挂载 CIFS 文件系统

如果想自动挂载 SMB/CIFS 共享（有关 SMB/CIFS 协议的更多信息，请参见第 20 章“Samba”），需要修改映射文件的语法。在 option 字段中添加 `-fstype=cifs`，并在共享位置前面加上一个冒号“:”。

```
mount point      -fstype=cifs      ://jupiter.com/export
```

A GNU 许可证

本附录包含 GNU 自由文档许可证版本 1.2。

GNU free documentation license

Copyright (C) 2000, 2001, 2002 Free Software Foundation, Inc. 51 Franklin St, Fifth Floor, Boston, MA 02110-1301 USA. Everyone is permitted to copy and distribute verbatim copies of this license document, but changing it is not allowed.

0. PREAMBLE

The purpose of this License is to make a manual, textbook, or other functional and useful document "free" in the sense of freedom: to assure everyone the effective freedom to copy and redistribute it, with or without modifying it, either commercially or non-commercially. Secondarily, this License preserves for the author and publisher a way to get credit for their work, while not being considered responsible for modifications made by others.

This License is a kind of "copyleft", which means that derivative works of the document must themselves be free in the same sense. It complements the GNU General Public License, which is a copyleft license designed for free software.

We have designed this License to use it for manuals for free software, because free software needs free documentation: a free program should come with manuals providing the same freedoms that the software does. But this License is not limited to software manuals; it can be used for any textual work, regardless of subject matter or whether it is published as a printed book. We recommend this License principally for works whose purpose is instruction or reference.

1. APPLICABILITY AND DEFINITIONS

This License applies to any manual or other work, in any medium, that contains a notice placed by the copyright holder saying it can be distributed under the terms of this License. Such a notice grants a world-wide, royalty-free license, unlimited in duration, to use that work under the conditions stated herein. The "Document", below, refers to any such manual or work. Any member of the public is a licensee, and is addressed as "you". You accept the license if you copy, modify or distribute the work in a way requiring permission under copyright law.

A "Modified Version" of the Document means any work containing the Document or a portion of it, either copied verbatim, or with modifications and/or translated into another language.

A "Secondary Section" is a named appendix or a front-matter section of the Document that deals exclusively with the relationship of the publishers or authors of the Document to the Document's overall subject (or to related matters) and contains nothing that could fall directly within that overall subject. (Thus, if the Document is in part a textbook of mathematics, a Secondary Section may not explain any mathematics.) The relationship could be a matter of historical connection with the subject or with related matters, or of legal, commercial, philosophical, ethical or political position regarding them.

The "Invariant Sections" are certain Secondary Sections whose titles are designated, as being those of Invariant Sections, in the notice that says that the Document is released under this License. If a section does not fit the above definition of Secondary then it is not allowed to be designated as Invariant. The Document may contain zero Invariant Sections. If the Document does not identify any Invariant Sections then there are none.

The "Cover Texts" are certain short passages of text that are listed, as Front-Cover Texts or Back-Cover Texts, in the notice that says that the Document is released under this License. A Front-Cover Text may be at most 5 words, and a Back-Cover Text may be at most 25 words.

A "Transparent" copy of the Document means a machine-readable copy, represented in a format whose specification is available to the general public, that is suitable for revising the document straightforwardly with generic text editors or (for images composed of pixels) generic paint programs or (for drawings) some widely available drawing editor, and that is suitable for input to text formatters or for automatic translation to a variety of formats suitable for input to text formatters. A copy made in an otherwise Transparent file format whose markup, or absence of markup, has been arranged to thwart or discourage subsequent modification by readers is not Transparent. An image format is not Transparent if used for any substantial amount of text. A copy that is not "Transparent" is called "Opaque".

Examples of suitable formats for Transparent copies include plain ASCII without markup, Texinfo input format, LaTeX input format, SGML or XML using a publicly available DTD, and standard-conforming simple HTML, PostScript or PDF designed for human modification. Examples of transparent image formats include PNG, XCF and JPG. Opaque formats include proprietary formats that can be read and edited only by proprietary word processors, SGML or XML for which the DTD and/or processing tools are not generally available, and the machine-generated HTML, PostScript or PDF produced by some word processors for output purposes only.

The "Title Page" means, for a printed book, the title page itself, plus such following pages as are needed to hold, legibly, the material this License requires to appear in the title page. For works in formats which do not have any title page as such, "Title Page" means the text near the most prominent appearance of the work's title, preceding the beginning of the body of the text.

A section "Entitled XYZ" means a named sub-unit of the Document whose title either is precisely XYZ or contains XYZ in parentheses following text that translates XYZ in another language. (Here XYZ stands for a specific section name mentioned below, such as "Acknowledgements", "Dedications", "Endorsements", or "History".) To "Preserve the Title" of such a section when you modify the Document means that it remains a section "Entitled XYZ" according to this definition.

The Document may include Warranty Disclaimers next to the notice which states that this License applies to the Document. These Warranty Disclaimers are considered to be included by reference in this License, but only as regards disclaiming warranties: any other implication that these Warranty Disclaimers may have is void and has no effect on the meaning of this License.

2. VERBATIM COPYING

You may copy and distribute the Document in any medium, either commercially or non-commercially, provided that this License, the copyright notices, and the license notice saying this License applies to the Document are reproduced in all copies, and that you add no other conditions whatsoever to those of this License. You may not use technical measures to obstruct or control the reading or further copying of the copies you make or distribute. However, you may accept compensation in exchange for copies. If you distribute a large enough number of copies you must also follow the conditions in section 3.

You may also lend copies, under the same conditions stated above, and you may publicly display copies.

3. COPYING IN QUANTITY

If you publish printed copies (or copies in media that commonly have printed covers) of the Document, numbering more than 100, and the Document's license notice requires Cover Texts, you must enclose the copies in covers that carry, clearly and legibly, all these Cover Texts: Front-Cover Texts on the front cover, and Back-Cover Texts on the back cover. Both covers must also clearly and legibly identify you as the publisher of these copies. The front cover must present the full title with all words of the title equally prominent and visible. You may add other material on the covers in addition. Copying with changes limited to the covers, as long as they preserve the title of the Document and satisfy these conditions, can be treated as verbatim copying in other respects.

If the required texts for either cover are too voluminous to fit legibly, you should put the first ones listed (as many as fit reasonably) on the actual cover, and continue the rest onto adjacent pages.

If you publish or distribute Opaque copies of the Document numbering more than 100, you must either include a machine-readable Transparent copy along with each Opaque copy, or state in or with each Opaque copy a computer-network location from which the general network-using public has access to download using public-standard network protocols a complete Transparent copy of the Document, free of added material. If you use the latter option, you must take reasonably prudent steps, when you begin distribution of Opaque copies in quantity, to ensure that this Transparent copy will remain thus accessible at the stated location until at least one year after the last time you distribute an Opaque copy (directly or through your agents or retailers) of that edition to the public.

It is requested, but not required, that you contact the authors of the Document well before redistributing any large number of copies, to give them a chance to provide you with an updated version of the Document.

4. MODIFICATIONS

You may copy and distribute a Modified Version of the Document under the conditions of sections 2 and 3 above, provided that you release the Modified Version under precisely this License, with the Modified Version filling the role of the Document, thus licensing distribution and modification of the Modified Version to whoever possesses a copy of it. In addition, you must do these things in the Modified Version:

- **A.** Use in the Title Page (and on the covers, if any) a title distinct from that of the Document, and from those of previous versions (which should, if there were any, be listed in the History section of the Document). You may use the same title as a previous version if the original publisher of that version gives permission.
- **B.** List on the Title Page, as authors, one or more persons or entities responsible for authorship of the modifications in the Modified Version, together with at least five of the principal authors of the Document (all of its principal authors, if it has fewer than five), unless they release you from this requirement.
- **C.** State on the Title page the name of the publisher of the Modified Version, as the publisher.
- **D.** Preserve all the copyright notices of the Document.
- **E.** Add an appropriate copyright notice for your modifications adjacent to the other copyright notices.
- **F.** Include, immediately after the copyright notices, a license notice giving the public permission to use the Modified Version under the terms of this License, in the form shown in the Addendum below.
- **G.** Preserve in that license notice the full lists of Invariant Sections and required Cover Texts given in the Document's license notice.
- **H.** Include an unaltered copy of this License.
- **I.** Preserve the section Entitled "History", Preserve its Title, and add to it an item stating at least the title, year, new authors, and publisher of the Modified Version as given on the Title Page. If there is no section Entitled "History" in the Document, create one stating the title, year, authors, and publisher of the Document as given on its Title Page, then add an item describing the Modified Version as stated in the previous sentence.
- **J.** Preserve the network location, if any, given in the Document for public access to a Transparent copy of the Document, and likewise the network locations given in the Document for previous versions it was based on. These may be placed in the "History" section. You may omit a network location for a work that was published at least four years before the Document itself, or if the original publisher of the version it refers to gives permission.
- **K.** For any section Entitled "Acknowledgements" or "Dedications", Preserve the Title of the section, and preserve in the section all the substance and tone of each of the contributor acknowledgements and/or dedications given therein.
- **L.** Preserve all the Invariant Sections of the Document, unaltered in their text and in their titles. Section numbers or the equivalent are not considered part of the section titles.
- **M.** Delete any section Entitled "Endorsements". Such a section may not be included in the Modified Version.
- **N.** Do not retitle any existing section to be Entitled "Endorsements" or to conflict in title with any Invariant Section.
- **O.** Preserve any Warranty Disclaimers.

If the Modified Version includes new front-matter sections or appendices that qualify as Secondary Sections and contain no material copied from the Document, you may at your option designate some or all of these sections as invariant. To do this, add their titles to the list of Invariant Sections in the Modified Version's license notice. These titles must be distinct from any other section titles.

You may add a section Entitled "Endorsements", provided it contains nothing but endorsements of your Modified Version by various parties--for example, statements of peer review or that the text has been approved by an organization as the authoritative definition of a standard.

You may add a passage of up to five words as a Front-Cover Text, and a passage of up to 25 words as a Back-Cover Text, to the end of the list of Cover Texts in the Modified Version. Only one passage of Front-Cover Text and one of Back-Cover Text may be added by (or through arrangements made by) any one entity. If the Document already includes a cover text for the same cover, previously added by you or by arrangement made by the same entity you are acting on behalf of, you may not add another; but you may replace the old one, on explicit permission from the previous publisher that added the old one.

The author(s) and publisher(s) of the Document do not by this License give permission to use their names for publicity for or to assert or imply endorsement of any Modified Version.

5. COMBINING DOCUMENTS

You may combine the Document with other documents released under this License, under the terms defined in section 4 above for modified versions, provided that you include in the combination all of the Invariant Sections of all of the original documents, unmodified, and list them all as Invariant Sections of your combined work in its license notice, and that you preserve all their Warranty Disclaimers.

The combined work need only contain one copy of this License, and multiple identical Invariant Sections may be replaced with a single copy. If there are multiple Invariant Sections with the same name but different contents, make the title of each such section unique by adding at the end of it, in parentheses, the name of the original author or publisher of that section if known, or else a unique number. Make the same adjustment to the section titles in the list of Invariant Sections in the license notice of the combined work.

In the combination, you must combine any sections Entitled "History" in the various original documents, forming one section Entitled "History"; likewise combine any sections Entitled "Acknowledgements", and any sections Entitled "Dedications". You must delete all sections Entitled "Endorsements".

6. COLLECTIONS OF DOCUMENTS

You may make a collection consisting of the Document and other documents released under this License, and replace the individual copies of this License in the various documents with a single copy that is included in the collection, provided that you follow the rules of this License for verbatim copying of each of the documents in all other respects.

You may extract a single document from such a collection, and distribute it individually under this License, provided you insert a copy of this License into the extracted document, and follow this License in all other respects regarding verbatim copying of that document.

7. AGGREGATION WITH INDEPENDENT WORKS

A compilation of the Document or its derivatives with other separate and independent documents or works, in or on a volume of a storage or distribution medium, is called an "aggregate" if the copyright resulting from the compilation is not used to limit the legal rights of the compilation's users beyond what the individual works permit. When the Document is included in an aggregate, this License does not apply to the other works in the aggregate which are not themselves derivative works of the Document.

If the Cover Text requirement of section 3 is applicable to these copies of the Document, then if the Document is less than one half of the entire aggregate, the Document's Cover Texts may be placed on covers that bracket the Document within the aggregate, or the electronic equivalent of covers if the Document is in electronic form. Otherwise they must appear on printed covers that bracket the whole aggregate.

8. TRANSLATION

Translation is considered a kind of modification, so you may distribute translations of the Document under the terms of section 4. Replacing Invariant Sections with translations requires special permission from their copyright holders, but you may include translations of some or all Invariant Sections in addition to the original versions of these Invariant Sections. You may include a translation of this License, and all the license notices in the Document, and any Warranty Disclaimers, provided that you also include the original English version of this License and the original versions of those notices and disclaimers. In case of a disagreement between the translation and the original version of this License or a notice or disclaimer, the original version will prevail.

If a section in the Document is Entitled "Acknowledgements", "Dedications", or "History", the requirement (section 4) to Preserve its Title (section 1) will typically require changing the actual title.

9. TERMINATION

You may not copy, modify, sublicense, or distribute the Document except as expressly provided for under this License. Any other attempt to copy, modify, sublicense or distribute the Document is void, and will automatically terminate your rights under this License. However, parties who have received copies, or rights, from you under this License will not have their licenses terminated so long as such parties remain in full compliance.

10. FUTURE REVISIONS OF THIS LICENSE

The Free Software Foundation may publish new, revised versions of the GNU Free Documentation License from time to time. Such new versions will be similar in spirit to the present version, but may differ in detail to address new problems or concerns. See http://www.gnu.org/copyleft/.

Each version of the License is given a distinguishing version number. If the Document specifies that a particular numbered version of this License "or any later version" applies to it, you have the option of following the terms and conditions either of that specified version or of any later version that has been published (not as a draft) by the Free Software Foundation. If the Document does not specify a version number of this License, you may choose any version ever published (not as a draft) by the Free Software Foundation.

ADDENDUM: How to use this License for your documents

```
Copyright (c) YEAR YOUR NAME.
Permission is granted to copy, distribute and/or modify this document
under the terms of the GNU Free Documentation License, Version 1.2
or any later version published by the Free Software Foundation;
with no Invariant Sections, no Front-Cover Texts, and no Back-Cover Texts.
A copy of the license is included in the section entitled “GNU
Free Documentation License”.
```

If you have Invariant Sections, Front-Cover Texts and Back-Cover Texts, replace the “with...Texts.” line with this:

```
with the Invariant Sections being LIST THEIR TITLES, with the
Front-Cover Texts being LIST, and with the Back-Cover Texts being LIST.
```

If you have Invariant Sections without Cover Texts, or some other combination of the three, merge those two alternatives to suit the situation.

If your document contains nontrivial examples of program code, we recommend releasing these examples in parallel under your choice of free software license, such as the GNU General Public License, to permit their use in free software.