Red Hat Enterprise Linux 9

配置设备映射器多路径

使用设备映射器多路径功能

Last Updated: 2022-06-10

Red Hat Enterprise Linux 9 配置设备映射器多路径

使用设备映射器多路径功能

Enter your first name here. Enter your surname here.
Enter your organisation's name here. Enter your organisational division here.
Enter your email address here.

法律通告

摘要

这个文档提供如何在 Red Hat Enterprise Linux 9 中配置和管理设备映射器多路径（DM-Multipath）功能的步骤。

目录

让开源更具包容性

红帽致力于替换我们的代码、文档和 Web 属性中存在问题的语言。我们从这四个术语开始：master、slave、黑名单和白名单。由于此项工作十分艰巨，这些更改将在即将推出的几个发行版本中逐步实施。如需了解更多详细信息，请参阅 CTO Chris Wright 信息。

对红帽文档提供反馈

我们感谢您对文档提供反馈信息。请让我们了解如何改进文档。

- 关于特定内容的简单评论：
  1. 请确定您使用 *Multi-page HTML* 格式查看文档。另外，确定 **Feedback** 按钮出现在文档页的右上方。
  2. 用鼠标指针高亮显示您想评论的文本部分。
  3. 点在高亮文本上弹出的 **Add Feedback**。
  4. 按照显示的步骤操作。
- 要通过 Bugzilla 提交反馈，请创建一个新的 ticket：
  1. 进入 Bugzilla 网站。
  2. 在 Component 中选择 **Documentation**。
  3. 在 **Description** 中输入您要提供的信息。包括文档相关部分的链接。
  4. 点 **Submit Bug**。

第 1 章 设备映射器多路径概述

使用设备映射器多路径（DM 多路径），您可以将服务器节点和存储阵列间的多个 I/O 路径配置为单一设备。这些 I/O 路径是可包含独立电缆、交换机和控制器的物理存储区域网络(SAN)连接。多路径聚合了 I/O 路径并生成由聚合路径组成的新设备。

DM 多路径提供：

冗余

DM 多路径可在主动/被动（active/passive）配置中提供故障切换。在主动/被动配置中，对于 I/O，任何时候都只会使用一半的路径。如果 I/O 路径的任何元素（如电缆、交换机或控制器）出现故障，DM 多路径会切换到备用路径。

改进的性能

可将 DM 多路径配置为主动/主动模式，其中将 I/O 以轮循（round-robin）方式分布到所有路径中。在一些配置中，DM 多路径可以检测 I/O 路径中的载入，并动态重新平衡负载。

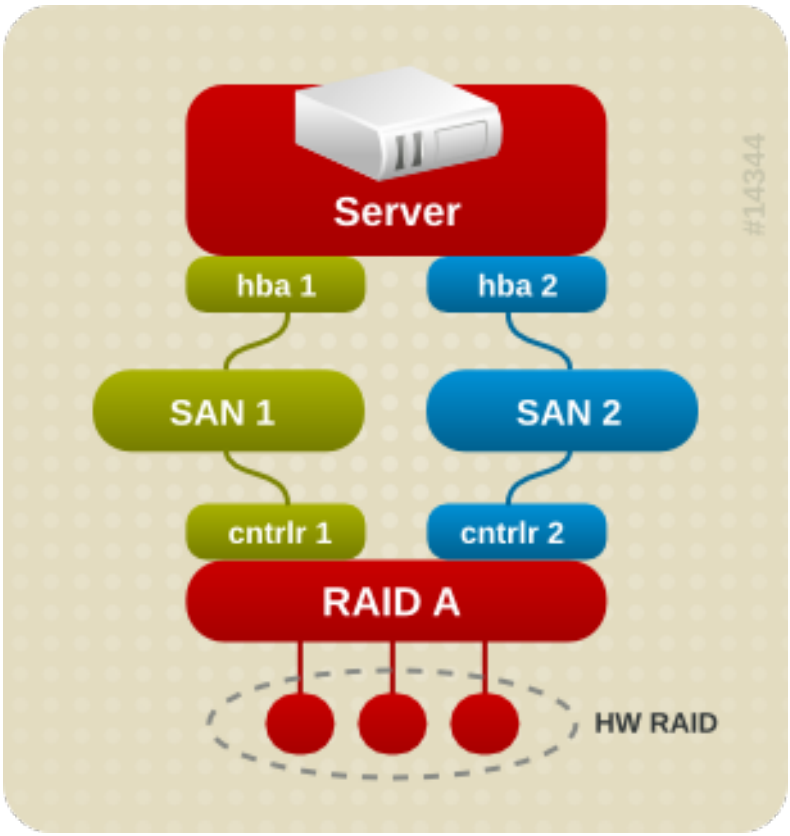
1.1. 带一个 RAID 设备的主动/被动多路径配置

在此配置中，服务器上有两个主机总线适配器(HBA)，两个 SAN 交换机和两个 RAID 控制器。以下是在这个配置中可能出现的故障：

- HBA 故障
- 光纤通道电缆失败
- SAN 交换机故障
- 阵列控制器端口故障

配置 DM 多路径后，任何这些点会导致 DM 多路径切换到备用 I/O 路径。以下镜像描述了服务器到 RAID 设备的两个 I/O 路径的配置。在这里，有一个 I/O 路径通过 **hba1**、**SAN1** 和 **cntrlr1**，第二个 I/O 路径则经过 Hba **2**、**SAN2** 和 **cntrlr2**。

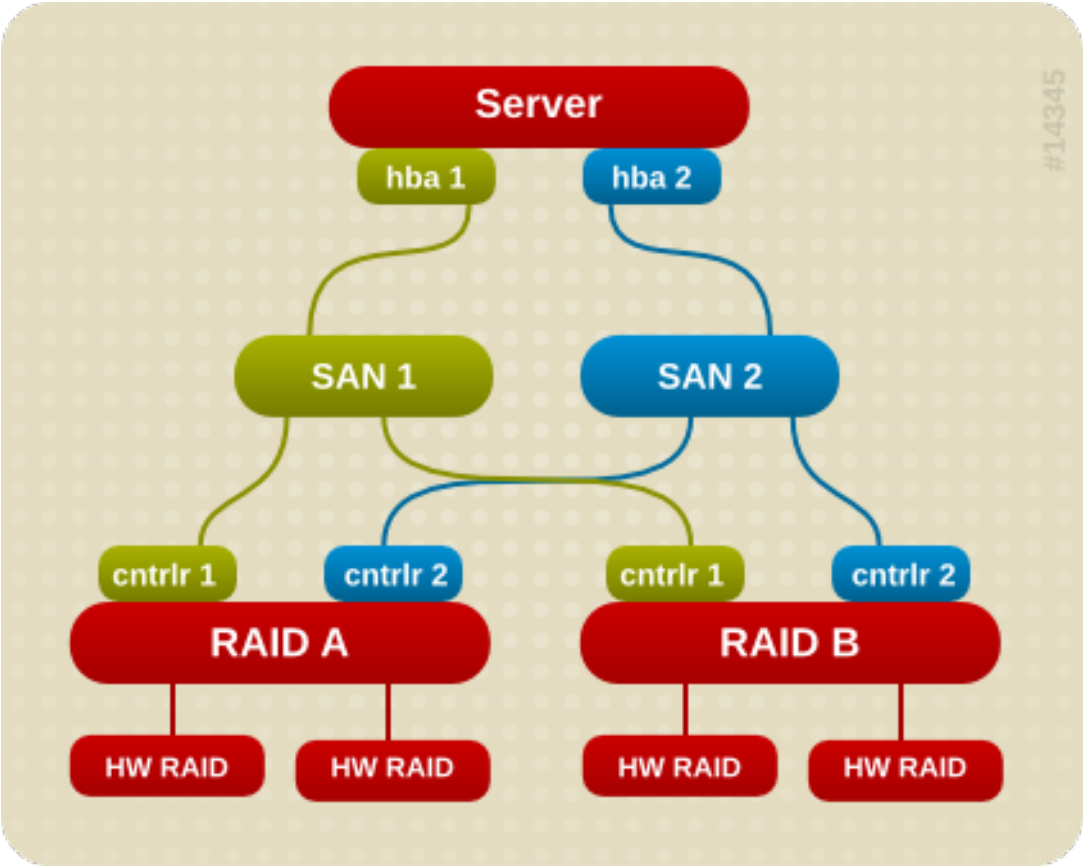
图 1.1. 带一个 RAID 设备的主动/被动多路径配置



1.2. 带两个 RAID 设备的主动/被动多路径配置

在此配置中，服务器中存在两个 HBA，两个 SAN 交换机，每个有两个 RAID 控制器。配置 DM 多路径后，在任意 RAID 设备的 I/O 路径点会导致 DM 多路径切换到该设备的备用 I/O 路径。以下镜像描述了每个 RAID 设备有两个 I/O 路径的配置。在这里，每个 RAID 设备有两个 I/O 路径。

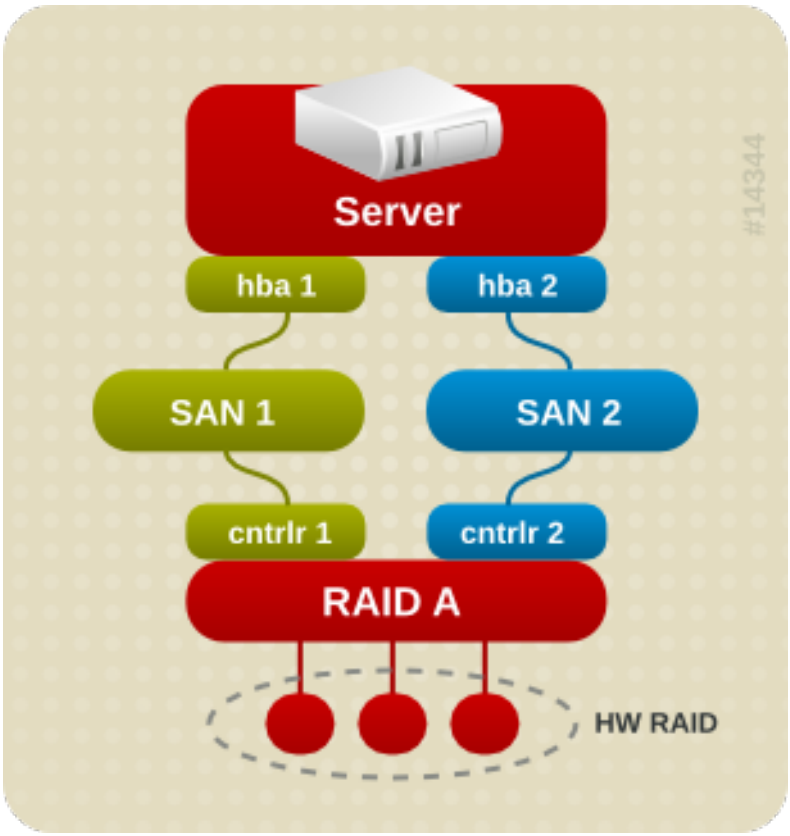
图 1.2. 带两个 RAID 设备的主动/被动多路径配置



1.3. 带一个 RAID 设备的主动/主动多路径配置

在此配置中，服务器中有两个 HBA、两个 SAN 交换机和两个 RAID 控制器。以下镜像描述了从服务器到存储设备的两个 I/O 路径的配置。在这里，可将 I/O 分布到这两个路径中。

图 1.3. 带一个 RAID 设备的主动/主动多路径配置



1.4. DM 多路径组件

下表描述了 DM 多路径组件。

表 1.1. DM 多路径的组件

组件	描述
dm_multipath 内核模块	为路径和路径组群重新路由 I/O 并支持故障切换。
mpathconf 工具	配置并启用设备映射器多路径。
multipath 命令	列出并配置多路径设备。每当添加块设备时，它也由 **udev** 执行，以确定该设备是否是多路径设备的一部分。
multipathd 守护进程	自动创建和删除多路径设备并监控路径；作为路径失败，可以更新多路径设备。允许对多路径设备进行交互式的修改。如果 **/etc/multipath.conf** 文件有任何更改，请重新加载该服务。
kpartx 命令	为设备中的分区创建设备映射器设备。当创建了多路径设备以便在其之上创建分区设备时，该命令将由 **udev** 自动执行。**kpartx** 命令在其自己的软件包中提供，但 **device-mapper-multipath** 软件包依赖于它。

mpathpersist	在多路径设备中设置 **SCSI-3** 持久预留。该命令的工作方式类似于 **sg_persist** 对不是多路径的 SCSI 设备工作，但它处理多路径设备上所有路径上的持久预留。它与 **多路径** 协调，以确保在稍后添加的路径上正确设置保留。要使用此功能，必须在 **/etc/multipath.conf** 文件中定义 **reservation_key** 属性。否则 **multipathd** 守护进程将不会检查新发现的路径或恢复的路径。

1.5. MULTIPATH 命令

multipath 命令用于检测和组合到设备的多个路径。它提供不同的选项来管理您的多路径设备。

下表描述了您可能会用到的 **multipath** 命令的一些选项。

表 1.2. 有用的 多路径 命令选项

选项	描述
-l	显示来自 **sysfs** 和设备映射器的当前多路径配置。
-ll	显示来自 **sysfs**、设备映射器以及系统中所有其他可用组件的当前多路径配置。
-f *设备*	删除命名的多路径设备。
-F	删除所有未使用的多路径设备。
-w *设备*	从 **wwid s** 文件中删除指定设备的 wwid。
-W	重置 **wwids** 文件，使其只包含当前的多路径设备。

1.6. MULTIPATH 命令输出

当您创建、修改或者列出多路径设备时，您会看到当前设备设置的显示。格式如下。

- 对于每个多路径设备：

```
action_if_any: alias (wwid_if_different_from_alias) dm_device_name_if_known
vendor,product size=size features='features' hwhandler='hardware_handler'
wp=write_permission_if_known
```

- 对于每个路径组：

```
-+- policy='scheduling_policy' prio=prio_if_known status=path_group_status_if_known
```

- 对于每个路径：

```
  `- host:channel:id:lun devnode major:minor dm_status_if_known path_status online_status
```

例如： multipath 命令的输出结果可能如下：

```
3600d0230000000000e13955cc3757800 dm-1 WINSYS,SF2372
size=269G features='0' hwhandler='0' wp=rw
|-+- policy='round-robin 0' prio=1 status=active
| `- 6:0:0:0 sdb 8:16  active ready  running
`-+- policy='round-robin 0' prio=1 status=enabled
  `- 7:0:0:0 sdf 8:80  active ready  running
```

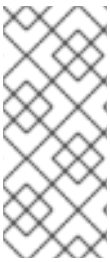
如果路径已启动并准备好进行 I/O，则路径的状态为 **ready** 或 **ghost**。如果路径停机，其状态为 **faulty** 或 **shaky**。路径状态由 **multipathd** 守护进程根据 **/etc/multipath.conf** 文件中定义的轮询间隔定期进行更新。

其他可能的路径状态值如下。

- **I/o pending** ：检查程序正在主动检查这个路径，状态将很快更新。
- **I/o timeout** ：这与 **故障** 相同。它让用户知道检查程序在超时时间前没有返回成功或失败。
- **删除** ：路径已经从系统中删除，并将很快从多路径设备中删除。它被处理与 **故障** 相同。
- **通配符** ：因为内部错误或配置问题，多路径无法运行路径检查器。**这与 错误 大致相同，但多路径会跳过路径上的多个操作。**
- **取消选中 ：路径检查程序没有在此路径上运行，因为它只是发现它，它没有分配的路径检查器，或者路径检查程序遇到了错误。这与 通配符 相同。**
- **延迟 ：路径检查程序返回路径的启动，但多路径会延迟路径的重新状态，因为路径最近多次失败，多路径已被配置为延迟路径。**

对于内核而言，dm 状态与路径状态类似。active dm 状态涵盖了 ready 和 ghost 路径状态。待处理的路径状态没有对等的 dm 状态。所有其他路径状态都会映射到 失败的 dm 状态。dm 状态将保持其当前状态，直到路径检查程序完成为止。

在线_status 的可能值 正在运行 和 离线。离线状态 表示这个 SCSI 设备已被禁用。



注意

当您创建或修改多路径设备时，多路径会输出该设备配置。但是，某些功能（例如，写入权限和其他功能信息）可能未知。输出和您在创建或修改过程中选择的功能之间可能会有区别。这是正常的行为。创建后列出设备以查看正确的状态。

1.7. 显示多路径配置

您可以使用 -l 和 multipath 命令来显示当前的多路径配置。l 选项显示从 sysfs 和设备映射器中的信息收集的多路径拓扑。t 选项显示 - l 选项显示的信息，除了系统所有其他可用组件之外。

在显示多路径配置时，您可以使用 multipath 命令的 -v 选项指定详细程度。指定 -v0 时不产生任何输出。指定 -v1 仅输出创建或更新的多路径名称，然后您可以传递给其他工具，如 kpartx。指定 -v2 会输出所有检测到的路径、多路径和设备映射。如需更多详细信息，还可以指定 -v3、-v 4 或 -v5。

以下示例显示了 multipath -l 命令的输出。

```
# multipath -l
3600d0230000000000e13955cc3757800 dm-1 WINSYS,SF2372
size=269G features='0' hwhandler='0' wp=rw
|-- policy='round-robin 0' prio=1 status=active
| `- 6:0:0:0 sdb 8:16  active ready  running
`-- policy='round-robin 0' prio=1 status=enabled
  `- 7:0:0:0 sdf 8:80  active ready  running
```

以下示例显示了 multipath -ll 命令的输出。

```
# multipath -ll
3600d0230000000000e13955cc3757801 dm-10 WINSYS,SF2372
size=269G features='0' hwhandler='0' wp=rw
|-- policy='round-robin 0' prio=1 status=enabled
| `- 19:0:0:1 sdc 8:32  active ready  running
`-- policy='round-robin 0' prio=1 status=enabled
  `- 18:0:0:1 sdh 8:112 active ready  running
3600d0230000000000e13955cc3757803 dm-2 WINSYS,SF2372
size=125G features='0' hwhandler='0' wp=rw
`-+- policy='round-robin 0' prio=1 status=active
  |- 19:0:0:3 sde 8:64  active ready  running
  `- 18:0:0:3 sdj 8:144 active ready  running
```

1.8. 其它资源

- multipath(8) 和 multipathd(8) man page
- /etc/multipath.conf file

第 2 章 多路径设备

DM 多路径提供了一种逻辑地整理 I/O 路径的方法，方法是在基础设备上创建单一多路径设备。如果没有 DM 多路径，系统会将服务器节点中的每个路径都把一个存储控制器视为单独的设备，即使 I/O 路径将相同的服务器节点连接到同一存储控制器。

2.1. 多路径设备识别符

当新设备受 DM 多路径控制时，这些设备会在 /dev/mapper/ 和 /dev/ 目录中创建。

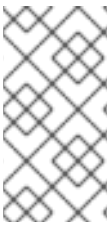
注意

任何格式为 /dev/dm-X 的设备都仅供内部使用，且绝不应该被管理员使用。

下面描述了多路径设备名称：

- 当 user_friendly_names 配置选项设置为 no 时，多路径设备的名称被设置为 World Wide Identifier(WWID)。默认情况下，多路径设备的名称被设置为它的 WWID。该设备名称应该是 /dev/mapper/WWID。它还在 /dev/ 目录中创建，名为 /dev/dm-X。

- 另外，您可以在 /etc/multipath.conf 文件中将 user_friendly_names 选项设置为 yes。这会将 multipath 部分中的 别名 设置为 mpathN 格式的节点唯一名称。该设备名称应该是 /dev/mapper/mpathN 和 /dev/dm-X。但不能保证，在所有使用多路径设备的节点中的设备名称都是一致的。同样，如果您在 /etc/multipath.conf 文件中设置了 alias 选项，该名称不会自动在集群中的所有节点中保持一致。



注意

如果您使用 LVM 在多路径设备中创建逻辑设备，这不应造成问题。为了使您的多路径设备名称在每个节点上一致，红帽建议禁用 user_friendly_names 选项。

例如：一个带有两个 HBA 的节点，通过一个没有区的 FC 交换机就可以看到四个设备：/dev/sda、/dev/sdb、/dev/sdc、/dev/sdc 和 /dev/sdd。DM 多路径会创建一个唯一 WWID 设备，它根据多路径配置将 I/O 重新路由到这四个底层设备。

除了 user_friendly_names 和 alias 选项外，多路径设备还具有其他属性。您可以通过在

/etc/multipath.conf 文件的 multipaths 部分中为该设备创建条目来修改特定多路径设备的这些属性。

其它资源

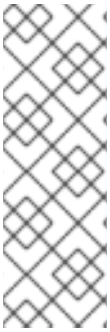
- multipath(8) 和 multipath.conf(8) man page
- /etc/multipath.conf file
- DM 多路径组件

2.2. 逻辑卷中的多路径设备

创建多路径设备后，您可以使用多路径设备名称，因为在创建逻辑卷管理器(LVM)物理卷时使用物理设备名称。例如，如果 /dev/mapper/mpatha 是多路径设备的名称，则 pvcreate /dev/mapper/mpatha 命令将 /dev/mapper/mpatha 标记为物理卷。

在创建 LVM 卷组时，您可以使用生成的 LVM 物理设备，就像使用其它 LVM 物理设备一样。

要过滤 /etc/lvm/lvm.conf 文件中的所有 sd 设备，请添加 filter = ["r/block/", "r/disk/", "r/sd./", "a/./"] 过滤器。



注意

如果您试图在配置的分区的整个设备中创建 LVM 物理卷，则 pvcreate 命令会失败。如果您不具体指定每个块设备，Anaconda 和 Kickstart 安装程序会创建空分区表。如果您要使用整个设备而不是创建分区，请从该设备中删除现有分区。您可以使用 kpartx -d device 命令和 fdisk 实用程序删除现有分区。如果您的系统有大于 2Tb 的块设备，请使用 parted 实用程序删除分区。

当您创建使用 主动/被动 多路径设备作为基础物理设备的 LVM 逻辑卷时，您可以选择在 /etc/lvm/lvm.conf 文件中包含过滤器，以排除多路径设备下的磁盘。这是因为如果阵列在收到 I/O 时自动更改被动路径，则当没有过滤这些设备时，多路径都会在 LVM 扫描被动路径时进行故障转移。

内核通过自动检测要使用的正确硬件处理程序来更改主动/被动状态。对于需要干预以改变其状态的主动/被动路径，多路径会自动使用这个硬件处理器根据需要进行操作。如果内核没有自动检测要使用的正确

硬件处理程序，您可以使用"hardware_handler"选项配置 multipath.conf 文件中要使用的硬件处理程序。对于需要命令主动制作 被动路径的主动/ 被动阵列，LVM 会在发生这种情况时打印警告信息。

根据您的配置，LVM 可能会打印以下任何信息：

- LUN 未就绪：

```
end_request: I/O error, dev sdc, sector 0
sd 0:0:0:3: Device not ready: <6>: Current: sense key: Not Ready
    Add. Sense: Logical unit not ready, manual intervention required
```

- 读失败：

```
/dev/sde: read failed after 0 of 4096 at 0: Input/output error
```

以下是上述错误的原因：

- 在为机器提供主动/被动路径的存储设备中设置多路径。
- 路径是直接访问的，而不是通过多路径设备访问。

其它资源

- lvm.conf man page
- DM 多路径组件

第 3 章 配置 DM 多路径

您可以使用 mpathconf 工具设置 DM 多路径。这个工具会根据以下情况创建或编辑 /etc/multipath.conf 多路径配置文件：

- 如果 /etc/multipath.conf 文件已存在，则 mpathconf 实用程序将编辑该文件。
- 如果 /etc/multipath.conf 文件不存在，则 mpathconf 实用程序将从头开始创建 /etc/multipath.conf 文件。

3.1. 检查 DEVICE-MAPPER-MULTIPATH 软件包

在您的系统中设置 DM 多路径前，请确定您的系统是最新的，并包含 device-mapper-multipath 软件包。

流程

1. 检查您的系统是否包含 device-mapper-multipath 软件包：

```
# rpm -q device-mapper-multipath
device-mapper-multipath-current-package-version
```

如果您的系统没有包括这个软件包，它会输出以下内容：

```
package device-mapper-multipath is not installed
```

2. 如果您的系统没有包括这个软件包，请运行以下命令安装它：

```
# {PackageManager} install device-mapper-multipath
```

3.2. 为基本故障切换配置设置 DM 多路径

如果您需要在启动 multipathd 守护进程前编辑 /etc/multipath.conf 文件，请使用以下步骤为基本故障切换配置设置 DM 多路径。

流程

1. 启用多路径配置文件：

```
# mpathconf --enable
```

2. 如果需要，请编辑 /etc/multipath.conf 文件。DM 多路径的默认设置被编译到系统中，不需要在 /etc/multipath.conf 文件中明确设置。

path_grouping_policy 的默认值被设置为 failover，因此，您不需要编辑 /etc/multipath.conf 文件。

配置文件的初始默认部分配置您的系统，以便多路径设备的名称格式为 /dev/mapper/mpathn；如果没有此设置，则多路径设备的名称将别名化为该设备的 WWID。如果您不想使用用户友好的名称，您可以输入以下命令：

```
# mpathconf --enable --user_friendly_names n
```

如果您需要在启动 multipath 守护进程后编辑多路径配置文件，则必须执行 systemctl reload multipathd.service 命令以使更改生效。

3. 保存配置文件并退出编辑器。

4. 启动 multipath 守护进程并创建多路径设备：

```
# systemctl start multipathd.service
```

注意

如果您删除了 device-mapper-multipath 软件包，则不会删除 /etc/multipath.conf 文件，或者 /etc/multipath 目录中的任何文件，因为该目录可以包含仅限于当前列出的文件。您可能需要在以后的 device-mapper-multipath 软件包安装中手动删除这些文件。

3.3. 在生成多路径设备时忽略本地磁盘

有些机器在其内部磁盘中使用本地 SCSI 卡，我们不建议在这些设备中使用 DM 多路径。如果将 find_multipaths 配置参数设置为 on，则不必在这些设备上禁用多路径。

如果您没有将 **find_multipaths** 配置参数设置为 **on**，您可以使用以下步骤修改 **DM** 多路径配置文件，以便在配置多路径时忽略本地磁盘。

流程

1. 确定哪些磁盘是内部磁盘。在这些示例中，**/dev/sda** 是内部磁盘：

   - 显示现有的多路径设备：

     ```
     # multipath -v2 -l

     SIBM-ESXSST336732LC____F3ET0EP0Q000072428BX1 dm-2 WINSYS,SF2372
     size=33 GB features="0" hwhandler="0" wp=rw
     `-+- policy='round-robin 0' prio=0 status=active
       |- 0:0:0:0 sda 8:0 active undef running
     ```

   - 显示 **DM** 多路径可能创建的附加多路径设备：

     ```
     # multipath -v2 -d

     : SIBM-ESXSST336732LC____F3ET0EP0Q000072428BX1 undef WINSYS,SF2372
     size=33 GB features="0" hwhandler="0" wp=undef
     `-+- policy='round-robin 0' prio=1 status=undef
       |- 0:0:0:0 sda 8:0  undef ready running
     ```

2. 编辑 **/etc/multipath.conf** 文件的 **blacklist** 部分，使其包含此设备。

   使用它的 **WWID** 属性识别该设备。虽然您可以使用 **devnode** 类型识别 **sda** 设备，但这不是一个安全的步骤，因为重新引导时 **/dev/sda** 无法保证相同。

   在上例中，**/dev/sda** 设备的 **WWID** 是 **SIBM-ESXSST336732LC___F3ET0Q000072428BX1**。要忽略这个设备，请在 **/etc/multipath.conf** 文件中包含以下内容：

   ```
   blacklist {
       wwid SIBM-ESXSST336732LC____F3ET0EP0Q000072428BX1
   }
   ```

3. 通过运行以下命令之一修改多路径配置文件后，验证 **/etc/multipath.conf** 文件：

- 要显示任何配置错误，请运行：

```
# multipath -t > /dev/null
```

- 要显示使用添加的更改显示新配置，请运行：

```
# multipath -t
```

4. 重新载入 /etc/multipath.conf 文件并重新配置 multipathd 守护进程以使更改生效：

```
# service multipathd reload
```

其它资源

- multipath.conf(5) 手册页

3.4. 配置附加存储设备

默认情况下，DM 多路径包括对支持最常见存储阵列的支持，该存储阵列支持 DM 多路径。

流程

- 查看默认配置值，包括支持的设备：

```
# multipathd show config
# multipath -t
```

- 可选：要添加默认不支持的附加存储设备，请编辑 /etc/multipath.conf 文件并插入适当的设备信息。

以下示例 ilustrates 如何添加 HP Open-V 系列的信息。这会将设备设置为队列，并在所有路径失败后每个重试 12 次重试，每个重试 5 秒。

```
devices {
        device {
                vendor "HP"
                product "OPEN-V"
```

```
            no_path_retry 12
        }
}
```

3.5. 在 INITRAMFS 文件系统中设置多路径

您可以在 initramfs 文件系统中设置多路径。如果您不使用需要多路径的设备，则不需要设置它，直到引导离开 initramfs 文件系统为止。

先决条件

- 您已在系统中配置了 DM 多路径。

流程

- 运行以下命令，使用多路径配置文件重建 initramfs 文件系统：

```
# dracut --force --add multipath
```

如果您从 initramfs 文件系统运行多路径并对多路径配置文件进行任何更改，则必须重建 initramfs 文件系统以使更改生效。

第 4 章 在 NVME 设备中启用多路径

您可以通过光纤传输（如光纤通道(FC)）连接到您的系统的多路径 NVMe 设备。您可以在多个多路径解决方案之间进行选择。

4.1. 本地 NVME 多路径和 DM 多路径

NVMe 设备支持原生多路径功能。当在 NVMe 中配置多路径时，您可以在标准 DM 多路径和原生 NVMe 多路径之间进行选择。

DM 多路径和原生 NVMe 多路径都支持 NVMe 设备的 Asymmetric Namespace Access(ANA)多路径方案。ANA 识别目标与发起方之间的优化路径并提高性能。

当启用原生 NVMe 多路径时，它会全局地应用于所有 NVMe 设备。它可以提供更高的性能，但不包含 DM 多路径提供的所有功能。例如，原生 NVMe 多路径只支持 故障切换（failover）和循环（round-robin）路径选择方法。

默认情况下，Red Hat Enterprise Linux 9 中启用了 NVMe 多路径，也是推荐的多路径解决方案。

4.2. 启用原生 NVME 多路径

此流程使用原生 NVMe 多路径解决方案在连接的 NVMe 设备中启用多路径。

先决条件

- NVMe 设备连接到您的系统。

  有关通过光纤传输连接 NVMe 的详情请参考 NVMe over fabric 设备概述。

步骤

1. 检查内核中是否启用了原生 NVMe 多路径：

```
# cat /sys/module/nvme_core/parameters/multipath
```

这个命令显示以下之一：

N

禁用原生 NVMe 多路径。

Y

启用原生 NVMe 多路径。

2. 如果禁用原生 NVMe 多路径，使用以下方法之一启用它：

- 使用内核选项：

  i. 在内核命令行中添加 nvme_core.multipath=Y 选项：

  ```
  # grubby --update-kernel=ALL --args="nvme_core.multipath=Y"
  ```

  ii. 在 64 位 IBM Z 构架中更新引导菜单：

  ```
  # zipl
  ```

  iii. 重启系统。

- 使用内核模块配置文件：

  i. 使用以下内容创建 /etc/modprobe.d/nvme_core.conf 配置文件：

  ```
  options nvme_core multipath=Y
  ```

  ii. 备份 initramfs 文件系统：

  ```
  # cp /boot/initramfs-$(uname -r).img \
      /boot/initramfs-$(uname -r).bak.$(date +%m-%d-%H%M%S).img
  ```

iii. **重建 initramfs 文件系统：**

```
# dracut --force --verbose
```

iv. **重启系统：**

3. **可选： 在运行的系统中，更改 NVMe 设备中的 I/O 策略，以便在所有可用路径中分发 I/O：**

```
# echo "round-robin" > /sys/class/nvme-subsystem/nvme-subsys0/iopolicy
```

4. **可选：使用 udev 规则永久设置 I/O 策略。使用以下内容创建 /etc/udev/rules.d/71-nvme-io-policy.rules 文件：**

```
ACTION=="add|change", SUBSYSTEM=="nvme-subsystem", ATTR{iopolicy}="round-robin"
```

验证

1. **检查您的系统是否识别 NVMe 设备：**

```
# nvme list

Node             SN                   Model                                    Namespace Usage
Format           FW Rev
---------------- -------------------- ---------------------------------------- --------- --------------------------
- ---------------- --------
/dev/nvme0n1     a34c4f3a0d6f5cec     Linux                                    1         250.06  GB /
250.06  GB    512   B +  0 B   4.18.0-2
/dev/nvme0n2     a34c4f3a0d6f5cec     Linux                                    2         250.06  GB /
250.06  GB    512   B +  0 B   4.18.0-2
```

2. **列出所有连接的 NVMe 子系统：**

```
# nvme list-subsys

nvme-subsys0 - NQN=testnqn
\
 +- nvme0 fc traddr=nn-0x20000090fadd597a:pn-0x10000090fadd597a host_traddr=nn-
0x20000090fac7e1dd:pn-0x10000090fac7e1dd live
 +- nvme1 fc traddr=nn-0x20000090fadd5979:pn-0x10000090fadd5979 host_traddr=nn-
0x20000090fac7e1dd:pn-0x10000090fac7e1dd live
 +- nvme2 fc traddr=nn-0x20000090fadd5979:pn-0x10000090fadd5979 host_traddr=nn-
```

```
0x20000090fac7e1de:pn-0x10000090fac7e1de live
 +- nvme3 fc traddr=nn-0x20000090fadd597a:pn-0x10000090fadd597a host_traddr=nn-
0x20000090fac7e1de:pn-0x10000090fac7e1de live
```

检查活动传输类型。例如，**nvme0 fc** 表示设备通过光纤通道传输连接，**nvme tcp** 则表示设备通过 **TCP** 连接。

3. 如果您编辑了内核选项，请检查内核命令行中是否启用了原生 **NVMe** 多路径：

```
# cat /proc/cmdline

BOOT_IMAGE=[...] nvme_core.multipath=Y
```

4. 检查 **DM** 多路径报告了 **NVMe** 命名空间为，例如：**nvme0c0c0n1** 到 **nvme0c3n1**，而 ***不是***，例如： **nvme0n1** 到 **nvme3n1** ：

```
# multipath -e -ll | grep -i nvme

uuid.8ef20f70-f7d3-4f67-8d84-1bb16b2bfe03 [nvme]:nvme0n1 NVMe,Linux,4.18.0-2
| `- 0:0:1    nvme0c0n1 0:0     n/a   optimized live
| `- 0:1:1    nvme0c1n1 0:0     n/a   optimized live
| `- 0:2:1    nvme0c2n1 0:0     n/a   optimized live
  `- 0:3:1    nvme0c3n1 0:0     n/a   optimized live

uuid.44c782b4-4e72-4d9e-bc39-c7be0a409f22 [nvme]:nvme0n2 NVMe,Linux,4.18.0-2
| `- 0:0:1    nvme0c0n1 0:0     n/a   optimized live
| `- 0:1:1    nvme0c1n1 0:0     n/a   optimized live
| `- 0:2:1    nvme0c2n1 0:0     n/a   optimized live
  `- 0:3:1    nvme0c3n1 0:0     n/a   optimized live
```

5. 如果您更改了 **I/O** 策略，请检查 **round-robin** 是 **NVMe** 设备中的活跃 **I/O** 策略：

```
# cat /sys/class/nvme-subsystem/nvme-subsys0/iopolicy

round-robin
```

其他资源

- 配置内核命令行参数

4.3. 在 NVME 设备中启用 DM 多路径

这个过程使用 DM 多路径解决方案在连接的 NVMe 设备中启用多路径。

先决条件

- NVMe 设备连接到您的系统。

  有关通过光纤传输连接 NVMe 的详情请参考 NVMe over fabric 设备概述。

步骤

1. 检查是否禁用了原生 NVMe 多路径：

```
# cat /sys/module/nvme_core/parameters/multipath
```

   这个命令显示以下之一：

   N

   禁用原生 NVMe 多路径。

   Y

   启用原生 NVMe 多路径。

2. 如果启用了原生 NVMe 多路径，请禁用它：

   a. 在内核命令行中删除 nvme_core.multipath=Y 选项：

```
# grubby --update-kernel=ALL --remove-args="nvme_core.multipath=Y"
```

   b. 在 64 位 IBM Z 构架中更新引导菜单：

```
# zipl
```

   c. 如果存在，从 /etc/modprobe.d/nvme_core.conf 文件中删除 options nvme_core multipath=Y 行。

d. **重启系统。**

3. **确保启用了 DM 多路径：**

```
# systemctl enable --now multipathd.service
```

4. **在所有可用路径中分发 I/O。在 /etc/multipath.conf 文件中添加以下内容：**

```
device {
  vendor "NVME"
  product ".*"
  path_grouping_policy    group_by_prio
}
```

注意

当 DM 多路径管理 NVMe 设备时，/sys/class/nvme-subsys0/iopolicy 配置文件不会影响 I/O 分发。

5. **重新载入 multipathd 服务以应用配置更改：**

```
# multipath -r
```

6. **备份 initramfs 文件系统：**

```
# cp /boot/initramfs-$(uname -r).img \
    /boot/initramfs-$(uname -r).bak.$(date +%m-%d-%H%M%S).img
```

7. **重建 initramfs 文件系统：**

```
# dracut --force --verbose
```

验证

1. **检查您的系统是否识别 NVMe 设备：**

```
# nvme list
```

```
Node             SN                   Model                                    Namespace Usage
Format           FW Rev
---------------- -------------------- ---------------------------------------- --------- --------------------------
- ---------------- --------
/dev/nvme0n1     a34c4f3a0d6f5cec     Linux                                    1         250.06  GB /
250.06  GB    512   B +  0 B   4.18.0-2
/dev/nvme0n2     a34c4f3a0d6f5cec     Linux                                    2         250.06  GB /
250.06  GB    512   B +  0 B   4.18.0-2
/dev/nvme1n1     a34c4f3a0d6f5cec     Linux                                    1         250.06  GB /
250.06  GB    512   B +  0 B   4.18.0-2
/dev/nvme1n2     a34c4f3a0d6f5cec     Linux                                    2         250.06  GB /
250.06  GB    512   B +  0 B   4.18.0-2
/dev/nvme2n1     a34c4f3a0d6f5cec     Linux                                    1         250.06  GB /
250.06  GB    512   B +  0 B   4.18.0-2
/dev/nvme2n2     a34c4f3a0d6f5cec     Linux                                    2         250.06  GB /
250.06  GB    512   B +  0 B   4.18.0-2
/dev/nvme3n1     a34c4f3a0d6f5cec     Linux                                    1         250.06  GB /
250.06  GB    512   B +  0 B   4.18.0-2
/dev/nvme3n2     a34c4f3a0d6f5cec     Linux                                    2         250.06  GB /
250.06  GB    512   B +  0 B   4.18.0-2
```

2. **列出所有连接的 NVMe 子系统。检查命令报告为 nvme0n1 到 nvme3n2，而*不是*，例如：nvme0c0c0n1 到 nvme0c3n1 ：**

```
# nvme list-subsys

nvme-subsys0 - NQN=testnqn
\
 +- nvme0 fc traddr=nn-0x20000090fadd5979:pn-0x10000090fadd5979 host_traddr=nn-
0x20000090fac7e1dd:pn-0x10000090fac7e1dd live
 +- nvme1 fc traddr=nn-0x20000090fadd597a:pn-0x10000090fadd597a host_traddr=nn-
0x20000090fac7e1dd:pn-0x10000090fac7e1dd live
 +- nvme2 fc traddr=nn-0x20000090fadd5979:pn-0x10000090fadd5979 host_traddr=nn-
0x20000090fac7e1de:pn-0x10000090fac7e1de live
 +- nvme3 fc traddr=nn-0x20000090fadd597a:pn-0x10000090fadd597a host_traddr=nn-
0x20000090fac7e1de:pn-0x10000090fac7e1de live
```

```
# multipath -ll

mpathae (uuid.8ef20f70-f7d3-4f67-8d84-1bb16b2bfe03) dm-36 NVME,Linux
size=233G features='1 queue_if_no_path' hwhandler='0' wp=rw
`-+- policy='service-time 0' prio=50 status=active
  |- 0:1:1:1  nvme0n1 259:0   active ready running
  |- 1:2:1:1  nvme1n1 259:2   active ready running
  |- 2:3:1:1  nvme2n1 259:4   active ready running
  `- 3:4:1:1  nvme3n1 259:6   active ready running

mpathaf (uuid.44c782b4-4e72-4d9e-bc39-c7be0a409f22) dm-39 NVME,Linux
size=233G features='1 queue_if_no_path' hwhandler='0' wp=rw
`-+- policy='service-time 0' prio=50 status=active
  |- 0:1:2:2  nvme0n2 259:1   active ready running
```

```
|- 1:2:2:2  nvme1n2 259:3   active ready running
|- 2:3:2:2  nvme2n2 259:5   active ready running
`- 3:4:2:2  nvme3n2 259:7   active ready running
```

其他资源

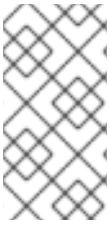
- 配置内核命令行参数
- 配置 DM 多路径.

第 5 章 修改 DM-MULTIPATH 配置文件

默认情况下，DM 多路径会为多数常见的多路径用例提供配置值。另外，DM 多路径包括对自己支持 DM 多路径的最常见存储阵列的支持。您可以通过编辑 /etc/multipath.conf 配置文件来覆盖 DM 多路径的默认配置值。如果需要，您还可以在配置文件中添加不支持的默认存储阵列。

有关默认配置值（包括支持的设备）的详情，请运行以下命令：

```
# multipathd show config
# multipath -t
```



注意

如果您从 initramfs 文件系统运行多路径并对多路径配置文件进行任何更改，则必须重建 initramfs 文件系统以使更改生效

在多路径配置文件中，您只需要指定配置所需的部分，或者从默认值中更改的小节。如果文件中没有与您的环境相关的部分，或者不需要覆盖默认值，您可以将其注释掉，因为它们位于初始文件中。

配置文件允许正则表达式描述语法。

5.1. 配置文件概述

多路径配置文件可分为以下几个部分：

黑名单

不视为多路径的特定设备列表。

blacklist_exceptions

根据 blacklist 部分的参数，列出其他将被忽略的多路径设备。

defaults

DM 多路径的常规默认设置。

multipaths

各个多路径设备特性的设置。这些值覆盖了在配置文件的、设备 和 defaults 部分中指定的内容。

devices

各个存储控制器的设置。这些值覆盖了在 配置文件的 defaults 部分中指定的内容。如果您使用默认不支持的存储阵列，您可能需要为阵列创建 devices 子部分。

overrides

适用于所有设备的设置。这些值覆盖了在配置文件的 devices 和 defaults 部分中指定的值。

当系统决定多路径设备的属性时，它会按照以下顺序检查 multipath.conf 文件中的单独部分的设置：

1. multipaths 部分
2. overrides 部分
3. devices 部分
4. defaults 部分

5.2. DM 多路径覆盖设备超时

restore_tmo sysfs 选项控制一个特定 iSCSI 设备的超时时间。以下选项全局覆盖 recovery_tmo 值：

- replacement_timeout 配置选项会全局覆盖所有 iSCSI 设备的 recovery_tmo 值。
- 对于由 DM 多路径管理的所有 iSCSI 设备，DM 多路径中的 fast_io_fail_tmo 选项会全局覆盖 recovery_tmo 值。

  DM 多路径中的 fast_io_fail_tmo 选项会覆盖光纤通道设备的 fast_io_fail_tmo 选项。

DM 多路径 fast_io_fail_tmo 选项优先于 replacement_timeout。红帽不推荐使用 replacement_timeout 在由 DM 多路径管理 的设备中覆盖 restore_tmo，因为当多路径服务重新加载

时，DM 多路径总是重置 restore _tmo。

5.3. 修改多路径配置文件默认设置

/etc/multipath.conf 配置文件包含一个 defaults 部分，该部分将 user_friendly_names 参数设置为 yes，如下所示。

```
defaults {
        user_friendly_names yes
}
```

这会覆盖 user_friendly_names 参数的默认值。multipath.conf 文件 的 defaults 部分中设置的默认值由 DM 多路径使用，除非被 multipath.conf 文件的设备、多路径或覆盖部分中指定的属性所覆盖。

流程

1. 查看 /etc/multipath.conf 配置文件，其中包含配置默认值模板：

```
#defaults {
#       polling_interval        10
#       path_selector           "round-robin 0"
#       path_grouping_policy    multibus
#       uid_attribute           ID_SERIAL
#       prio                    alua
#       path_checker            readsector0
#       rr_min_io               100
#       max_fds                 8192
#       rr_weight               priorities
#       failback                immediate
#       no_path_retry           fail
#       user_friendly_names     yes
#}
```

2. 覆盖任何配置参数的默认值。您可以从此模板将相关行复制到 defaults 部分，并取消注释它。

例如，要将 path_grouping_policy 参数覆盖为 multibus，而不是 故障转移 的默认值，请将模板中的相应行复制到配置文件的初始默认值部分，然后取消对它的注释，如下所示：

```
defaults {
        user_friendly_names     yes
        path_grouping_policy    multibus
}
```

3. 通过运行以下命令之一修改多路径配置文件后，验证 /etc/multipath.conf 文件：

   - 要显示任何配置错误，请运行：

     ```
     # multipath -t > /dev/null
     ```

   - 要显示使用添加的更改显示新配置，请运行：

     ```
     # multipath -t
     ```

4. 重新载入 /etc/multipath.conf 文件并重新配置 multipathd 守护进程以使更改生效：

   ```
   # service multipathd reload
   ```

其它资源

- multipath.conf(5) 和 multipathd(8) man page

5.4. 修改具体设备的多路径设置

在 multipath.conf 配置文件的 multipaths 部分中，您可以添加特定于单个多路径设备的配置，由强制 WWID 参数引用。

这些默认设置由 DM 多路径使用，并覆盖 multipath.conf 文件的 overrides、default 和 devices 部分设置的属性。multipaths 部分可能存在任意数量的多路径子部分。

流程

1. 修改特定多路径设备的 multipaths 部分。以下示例显示了在配置文件中为两个特定多路径设备指定的多路径属性：

   - 第一个设备的 WWID 为 3600508b4000156d70001200000b0000，符号链接名为 yellow。

- 示例中的第二个多路径设备的 WWID 为 1DEC_321816758474，符号链接名为。

在本例中，rr_weight 属性设置为 priorities。

```
multipaths {
        multipath {
                wwid                    3600508b4000156d70001200000b0000
                alias                   yellow
                path_grouping_policy    multibus
                path_selector           "round-robin 0"
                failback                manual
                rr_weight               priorities
                no_path_retry           5
        }
        multipath {
                wwid                    1DEC_321816758474
                alias                   red
                rr_weight               priorities
        }
}
```

2. 通过运行以下命令之一修改多路径配置文件后，验证 /etc/multipath.conf 文件：

   - 要显示任何配置错误，请运行：

     ```
     # multipath -t > /dev/null
     ```

   - 要显示使用添加的更改显示新配置，请运行：

     ```
     # multipath -t
     ```

3. 重新载入 /etc/multipath.conf 文件并重新配置 multipathd 守护进程以使更改生效：

   ```
   # service multipathd reload
   ```

其它资源

- multipath.conf(5) 手册页

5.5. 修改存储控制器的多路径设置

multipath.conf 配置文件的 devices 部分为独立的存储设备设置属性。这些属性可由 DM 多路径使用，除非被 多路径 或者 覆盖了 包含该设备的路径的 multipath.conf 文件中指定的属性所覆盖。这些属性覆盖 multipath.conf 文件的 defaults 部分中设置的属性。

流程

1. 查看默认配置值的信息，包括支持的设备：

```
# multipathd show config
# multipath -t
```

在多路径配置中，默认包括支持多路径的许多设备。

2. 可选：如果需要修改默认配置值，您可以通过在配置文件中包含覆盖这些值的设备的条目来覆盖默认值。您可以复制 multipathd show config 命令显示的设备的设备配置默认值，并覆盖您要更改的值。

3. 通过设置 供应商和 产品 参数，将默认自动配置的设备添加到配置文件的 devices 部分。打开 /sys/block/*device_name*/device/vendor 和 /sys/block/*device_name*/device/model 文件，其中 *device_name* 是多路径的设备，如下例所示：

```
# cat /sys/block/sda/device/vendor
WINSYS
# cat /sys/block/sda/device/model
SF2372
```

4. 可选： 根据您的具体设备指定附加参数：

主动/主动 设备

通常，在这种情况下不需要设置附加参数。如果需要，您可以将 path_grouping_policy 设置为 multibus。其他可能需要设置的参数为 no_path_retry 和 rr_min_io。

主动/被动 设备

如果它自动将 I/O 的路径切换到被动路径，您需要将检查程序功能更改为不会将 I/O 发送到路径路径，以测试其是否工作，否则您的设备会保持故障。这意味着，您已将 path_checker 设置为 tur，它适用于支持 Test unit Ready 命令的所有 SCSI 设备。

如果设备需要特殊命令来切换路径，则为多路径配置这个设备需要硬件处理器内核模块。当前可用的硬件处理器是 emc。如果您的设备不够，您可能无法为多路径配置设备。

以下示例显示了多路径配置文件中的 设备 条目：

```
# }
# device {
#  vendor   "COMPAQ  "
#  product   "MSA1000         "
#  path_grouping_policy multibus
#  path_checker  tur
#  rr_weight  priorities
# }
#}
```

5. 通过运行以下命令之一修改多路径配置文件后，验证 /etc/multipath.conf 文件：

   - 要显示任何配置错误，请运行：

     ```
     # multipath -t > /dev/null
     ```

   - 要显示使用添加的更改显示新配置，请运行：

     ```
     # multipath -t
     ```

6. 重新载入 /etc/multipath.conf 文件并重新配置 multipathd 守护进程以使更改生效：

   ```
   # service multipathd reload
   ```

其它资源

- multipath.conf(5) 和 multipathd(8) man page

5.6. 为所有设备设定多路径值

使用 multipath.conf 配置文件的 overrides 部分，您可以为所有设备设置配置值。这部分支持 multipath.conf 配置文件的 devices 和 defaults 部分支持的所有属性，这是 供应商、产品和 修订 以外的所有 设备 部分属性。

这些属性可由 DM 多路径为所有设备使用，除非被 multipath.conf 文件的 multipath.conf 文件的

multipaths 部分中指定的属性覆盖。这些属性覆盖 multipath.conf 文件的 devices 和 defaults 部分中设置的属性。

流程

1. **覆盖特定于设备的设置。例如，您可能希望所有设备都设置为 no_path_retry，使其 失败。当所有路径都失败时，使用以下命令关闭队列。这会覆盖任何特定于设备的设置。**

```
overrides {
        no_path_retry fail
}
```

2. **通过运行以下命令之一修改多路径配置文件后，验证 /etc/multipath.conf 文件：**

   - **要显示任何配置错误，请运行：**

```
# multipath -t > /dev/null
```

   - **要显示使用添加的更改显示新配置，请运行：**

```
# multipath -t
```

3. **重新载入 /etc/multipath.conf 文件并重新配置 multipathd 守护进程以使更改生效：**

```
# service multipathd reload
```

其它资源

- **multipath.conf(5) 手册页**

第 6 章 防止设备多路径

您可以将 DM 多路径配置为在配置多路径设备时忽略所选设备。DM 多路径不会将这些忽略的设备分组到多路径设备中。

6.1. DM 多路径为路径创建多路径设备的条件

DM 多路径有一组默认规则，用于决定是否为路径创建多路径设备还是忽略路径。您可以配置行为。

如果将 find_multipaths 配置参数设定为 off，则多路径总是会尝试为每个未明确禁用的路径创建一个多路径设备。如果在 上将 find_multipaths 配置参数设置为，则多路径会创建一个设备，只有在满足以下条件之一时：

- 至少有两个路径有相同的全局-Wide Identification(WWID)没有禁用。
- 您可以使用 multipath 命令指定设备来手动强制创建设备。
- 一个路径的 WWID 与之前创建的多路径设备相同，即使那个多路径设备目前还不存在。每当创建多路径设备时，多路径都会记住设备的 WWID，以便在看到该 WWID 的路径时立即自动创建该设备。这可让您让多路径自动选择到多路径设备的正确路径，而无需在其它设备中禁用多路径。

如果您之前使用 find_multipaths 参数创建了多路径设备，然后稍后将 参数设置为 上的，您可能需要从 /etc/multipath/wwids 文件中删除您不想作为多路径设备创建的 WWID。以下示例显示了示例 /etc/multipath/wwids 文件。WWID 用斜杠(/)括起：

```
# Multipath wwids, Version : 1.0
# NOTE: This file is automatically maintained by multipath and multipathd.
# You should not need to edit this file in normal circumstances.
#
# Valid WWIDs:
/3600d0230000000000e13955cc3757802/
/3600d0230000000000e13955cc3757801/
/3600d0230000000000e13955cc3757800/
/3600d02300069c9ce09d41c31f29d4c00/
/SWINSYS  SF2372        0E13955CC3757802/
/3600d0230000000000e13955cc3757803/
```

除了 上 和关闭 之外，您还可以将 find_multipaths 设置为以下值：

strict

多路径永远不会接受之前没有多路径的路径，因此不在 /etc/multipath/wwids 文件中。

smart

多路径会在出现时立即接受 udev 中的非禁用设备。如果 multipathd 没有在使用 find_multipaths_timeout 参数设置的超时中创建设备，它将在该设备中释放其声明。

查找_multipaths 的内置默认值为 off。但是，mpathconf 创建的默认 multipath.conf 文件将在 上将 find_multipaths 的值设置为。

当 find_multipaths 参数设置为 on 时，仅在带有您不想多路径的多个路径的设备中禁用多路径。因此，通常不需要在设备中禁用多路径。

如果您将之前创建的多路径设备添加到 黑名单 中，通过使用 -w 选项从 /etc/multipath/wwids 文件中删除该设备的 WWID 有助于避免与其他程序出现问题。例如，要从 /etc/multipath/wwids 文件中删除 WWID 3600d023000000e13954ed5f89300 设备 /dev/sdb。

- 使用设备名称删除多路径设备。

```
#multipath -w /dev/sdb
wwid '3600d0230000000000e13954ed5f89300' removed
```

- 使用设备的 WWID 删除多路径设备。

```
#multipath -w 3600d0230000000000e13954ed5f89300
wwid '3600d0230000000000e13954ed5f89300' removed
```

您也可以使用 -W 选项来更新 /etc/multipath/wwids 文件。这会将 /etc/multipath/wwids 文件重置为仅包含当前多路径设备的 WWID。要重置文件，请运行以下命令：

```
#multipath -W
successfully reset wwids
```

其它资源

- multipath.conf(5) 手册页

6.2. 在某些设备中禁用多路径的条件

您可以根据以下标准在设备中禁用多路径：

- WWID
- 设备名称
- 设备类型
- 属性
- 协议

对于每个设备，DM 多路径会按照以下顺序评估这些条件：

1. 属性
2. devnode
3. device
4. 协议
5. wwid

如果某个设备被任何上述条件被禁用，DM 多路径会排除它处理多路径，且不会评估后续标准。对于每个条件，如果设备同时匹配，则异常列表优先于禁用的设备列表。

注意

默认情况下，禁用了各种设备类型，即使您注释掉了配置文件的初始 黑名单 部分。

其它资源

- 为禁用多路径的设备添加例外

6.3. 使用 WWID 禁用多路径

您可以通过其全局识别(WWID)禁用独立设备上的多路径。

流程

1. 使用 wwid 条目禁用 /etc/multipath.conf 配置文件中的设备。

   以下示例显示，DM 多路径配置文件中禁用 WWID 为 26353900f02796769 的设备的行：

```
blacklist {
        wwid 26353900f02796769
}
```

2. 通过运行以下命令之一修改多路径配置文件后，验证 /etc/multipath.conf 文件：

   - 要显示任何配置错误，请运行：

```
# multipath -t > /dev/null
```

   - 要显示使用添加的更改显示新配置，请运行：

```
# multipath -t
```

3. 重新载入 /etc/multipath.conf 文件并重新配置 multipathd 守护进程以使更改生效：

```
# service multipathd reload
```

6.4. 使用设备名称禁用多路径

您可以使用设备名称在设备类型中禁用多路径，以便 DM 多路径不会将其分组到多路径设备中。

流程

1. 使用 devnode 条目禁用 /etc/multipath.conf 配置文件中的设备。

   下面的例子显示，DM 多路径配置文件中禁用所有 SCSI 设备的行，因为它也禁用所有 sd* 设备：

```
blacklist {
        devnode "^sd[a-z]"
}
```

   您可以使用 devnode 条目禁用单个设备，而不是禁用特定类型的所有设备。但不建议这样做，因为除非由 udev 规则静态映射，否则无法保证重启后特定设备的名称相同。例如：重启后，设备名称可以从 /dev/sda 改为 /dev/sdb。

   默认情况下，DM 多路径会禁用所有不是 SCSI、NVMe 或者 DASD 的设备，使用以下 devnode 条目：

```
blacklist {
        devnode "!^(sd[a-z]|dasd[a-z]|nvme[0-9])"
}
```

   这个条目禁用的设备通常不支持 DM 多路径。

2. 通过运行以下命令之一修改多路径配置文件后，验证 /etc/multipath.conf 文件：

   - 要显示任何配置错误，请运行：

```
# multipath -t > /dev/null
```

   - 要显示使用添加的更改显示新配置，请运行：

```
# multipath -t
```

3. 重新载入 /etc/multipath.conf 文件并重新配置 multipathd 守护进程以使更改生效：

```
# service multipathd reload
```

其它资源

- 为禁用多路径的设备添加例外

6.5. 根据设备类型禁用多路径

您可以使用 device 部分在设备中禁用多路径。

流程

1. 使用 device 部分，禁用 /etc/multipath.conf 配置文件中的设备。

   以下示例禁用所有 IBM DS4200 和 HP 设备的多路径：

```
blacklist {
        device {
                vendor  "IBM"
                product "3S42"       #DS4200 Product 10
        }
        device {
                vendor  "HP"
                product ".*"
        }
}
```

2. 通过运行以下命令之一修改多路径配置文件后，验证 /etc/multipath.conf 文件：

   - 要显示任何配置错误，请运行：

```
# multipath -t > /dev/null
```

   - 要显示使用添加的更改显示新配置，请运行：

```
# multipath -t
```

3. 重新载入 /etc/multipath.conf 文件并重新配置 multipathd 守护进程以使更改生效：

```
# service multipathd reload
```

6.6. 使用 UDEV 属性禁用多路径

您可以通过其 udev 属性参数禁用对设备的多路径。

流程

1. 使用 property 参数禁用 /etc/multipath.conf 配置文件中的设备。此参数是一个正则表达式字符串，与设备的 udev 环境变量名称匹配。

   以下示例禁用了所有使用 udev 属性 ID_ATA 的设备上的多路径：

```
blacklist {
        property "ID_ATA"
}
```

2. 通过运行以下命令之一修改多路径配置文件后，验证 /etc/multipath.conf 文件：

   - 要显示任何配置错误，请运行：

```
# multipath -t > /dev/null
```

   - 要显示使用添加的更改显示新配置，请运行：

```
# multipath -t
```

3. 重新载入 /etc/multipath.conf 文件并重新配置 multipathd 守护进程以使更改生效：

```
# service multipathd reload
```

6.7. 使用设备协议禁用多路径

您可以使用设备协议禁用设备中的多路径。

流程

1. **可选：查看路径使用的协议：**

```
# multipathd show paths format "%d %P"
```

2. **使用 protocol 部分禁用 /etc/multipath.conf 配置文件中的设备。**

   以下示例在带有未定义协议或未知 SCSI 传输类型的所有设备中禁用多路径：

```
blacklist {
        protocol "scsi:unspec"
        protocol "undef"
}
```

   DM 多路径识别以下协议字符串：

   - **scsi:fcp**
   - **scsi:spi**
   - **scsi:ssa**
   - **scsi:sbp**
   - **scsi:srp**
   - **SCSI:iscsi**
   - **scsi:sas**

- **scsi:adt**
- **scsi:ata**
- **scsi:unspec**
- **ccw**
- **cciss**
- **nvme**
- **undef**

3. **通过运行以下命令之一修改多路径配置文件后，验证 /etc/multipath.conf 文件：**

   - **要显示任何配置错误，请运行：**

     ```
     # multipath -t > /dev/null
     ```

   - **要显示使用添加的更改显示新配置，请运行：**

     ```
     # multipath -t
     ```

4. **重新载入 /etc/multipath.conf 文件并重新配置 multipathd 守护进程以使更改生效：**

   ```
   # service multipathd reload
   ```

6.8. 为禁用多路径的设备添加例外

您可以通过在当前禁用多路径的设备中添加例外来启用多路径。

先决条件

- 在某些设备中禁用多路径。

流程

1. 使用 /etc/multipath.conf 配置文件的 blacklist_exceptions 部分在设备上启用多路径。

当在配置文件的 blacklist_exceptions 部分中指定设备时，您必须使用与 黑名单 部分中指定的相同标准指定例外。例如： WWID 异常不适用于 devnode 条目禁用的设备，即使禁用的设备与该 WWID 关联。同样，devnode 例外仅适用于 devnode 条目，设备 例外则仅适用于设备条目。

例 6.1. WWID 异常

如果您有大量设备，且希望仅多路径 WWID 为 3600d023000000e13955cc3757803，而不是逐一禁用每个设备，您可以禁用所有这些设备，然后禁用所有这些设备，然后通过将以下几行添加到 /etc/multipath.conf 文件中来只启用其中一个。

```
blacklist {
        wwid ".*"
}

blacklist_exceptions {
        wwid "3600d0230000000000e13955cc3757803"
}
```

另外，您可以使用感叹号(!)来反转 黑名单 条目，该条目会禁用除指定 WWID 之外的所有设备：

```
blacklist {
        wwid "!3600d0230000000000e13955cc3757803"
}
```

例 6.2. udev 属性的例外

属性 参数的工作方式与其他 blacklist_exception 参数不同。property 参数的值必须与 udev 数据库中变量名称匹配。否则，设备会被禁用。使用这个参数，您可以在某些 SCSI 设备中禁用多路径，如 USB 盘和本地硬盘。

要只在可能进行多路径的 SCSI 设备中启用多路径，请将此参数设置为 (SCSI_IDENT_|ID_WWN)，如下例所示：

```
blacklist_exceptions {
        property "(SCSI_IDENT_|ID_WWN)"
}
```

2. **通过运行以下命令之一修改多路径配置文件后，验证 /etc/multipath.conf 文件：**

   - **要显示任何配置错误，请运行：**

     ```
     # multipath -t > /dev/null
     ```

   - **要显示使用添加的更改显示新配置，请运行：**

     ```
     # multipath -t
     ```

3. **重新载入 /etc/multipath.conf 文件并重新配置 multipathd 守护进程以使更改生效：**

   ```
   # service multipathd reload
   ```

第 7 章 管理多路径卷

以下是 DM 多路径提供的几个命令，您可以使用它们来管理多路径卷：

- multipath
- dmsetup
- multipathd

7.1. 重新定义在线多路径设备大小

如果您需要重新定义在线多路径设备的大小，请使用以下步骤。

流程

1. 重新定义您的物理设备大小。
2. 执行以下命令查找逻辑单元号(LUN)的路径：

```
# multipath -l
```

3. 重新定义您的路径大小。对于 SCSI 设备，在 重新扫描 设备的重新扫描文件中写入 1 会导致 SCSI 驱动程序重新扫描，如下所示：

```
# echo 1 > /sys/block/path_device/device/rescan
```

请确定您为每个路径设备运行这个命令。例如：如果您的路径设备是 sda、sdb、sde 和 sdf，则您要运行以下命令：

```
# echo 1 > /sys/block/sda/device/rescan
# echo 1 > /sys/block/sdb/device/rescan
# echo 1 > /sys/block/sde/device/rescan
# echo 1 > /sys/block/sdf/device/rescan
```

4. 重新定义多路径设备大小：

```
# multipathd resize map multipath_device
```

5. 重新定义文件系统大小（假设没有使用 LVM 或者 DOS 分区）：

```
# resize2fs /dev/mapper/mpatha
```

7.2. 将 ROOT 文件系统从单一路径设备移动到多路径设备中

如果您在单一路径设备中安装了系统，之后在 root 文件系统中添加另一个路径，则需要将您的根文件系统移到多路径设备。有关从单一路径移动到多路径设备的详情，请查看以下步骤。

先决条件

- 已安装 device-mapper-multipath 软件包。

流程

1. 创建 /etc/multipath.conf 配置文件，加载 multipath 模块并启用 multipathd systemd 服务：

```
# yum install device-mapper-multipath
```

流程

1. 执行以下命令，以创建 /etc/multipath.conf 配置文件，载入 multipath 模块，并将 multipathd 的 chkconfig 设置为 on:

2. 如果 find_multipaths 配置参数未设置为 yes，请编辑 /etc/multipath.conf 文件的 blacklist 和 blacklist_exceptions 部分，如 多路径中的 Preventing 设备 所述。

3. 要让多路径在发现 root 设备后马上构建多路径设备，请输入以下命令。此命令还可确保 find_multipaths 允许设备，即使它只有一个路径。

```
# multipath -a root_devname
```

例如，如果 root 设备是 /dev/sdb，请输入以下命令。

```
# multipath -a /dev/sdb
wwid '3600d02300069c9ce09d41c4ac9c53200' added
```

4. 执行 multipath 命令并搜索以下格式行的输出，确认您的配置文件设置是否正确。这表示该命令创建多路径设备失败。

```
date  wwid: ignoring map
```

例如：如果设备的 WWID 是 *3600d02300069c9ce09d41c4ac9c53200*，您将在输出中看到一行，如下所示：

```
# multipath
Oct 21 09:37:19 | 3600d02300069c9ce09d41c4ac9c53200: ignoring map
```

5. 使用 多路径 重建 initramfs 文件系统：

```
# dracut --force -H --add multipath
```

6. 关闭机器。

7. 引导机器。

8. 使其他路径对机器可见。

验证步骤

- 运行以下命令，检查多路径设备是否已创建：

```
# multipath -l | grep 3600d02300069c9ce09d41c4ac9c53200
mpatha (3600d02300069c9ce09d41c4ac9c53200) dm-0 3PARdata,VV
```

7.3. 将 SWAP 文件系统从单一路径设备移动到多路径设备中

默认情况下将 swap 设备设定为逻辑卷。只要您在构成逻辑卷的物理卷中设置了多路径，就不需要将其

配置为多路径设备。如果您的 swap 设备不是 LVM 卷，且使用设备名称挂载，您可能需要编辑 /etc/fstab 文件以切换到适当的多路径设备名称。

流程

1. 将设备的 WWID 添加到 /etc/multipath/wwids 文件中：

```
# multipath -a swap_devname
```

例如，如果 root 设备是 /dev/sdb，请输入以下命令。

```
# multipath -a /dev/sdb
wwid '3600d02300069c9ce09d41c4ac9c53200' added
```

2. 执行 multipath 命令并搜索以下格式行的输出，确认您的配置文件设置是否正确：

```
date  wwid: ignoring map
```

这表示该命令创建多路径设备失败。

例如：如果设备的 WWID 是 3600d02300069c9ce09d41c4ac9c53200，您将在输出中看到一行，如下所示：

```
# multipath
Oct 21 09:37:19 | 3600d02300069c9ce09d41c4ac9c53200: ignoring map
```

3. 在 /etc/multipath.conf 文件中为交换设备设置别名：

```
multipaths {
   multipath {
      wwid WWID_of_swap_device
      alias swapdev
   }
}
```

4. 编辑 /etc/fstab 文件，并使用多路径设备替换到 root 设备的旧设备路径。

例如，如果您在 /etc/fstab 文件中有以下条目：

```
/dev/sdb2 swap                  swap    defaults        0 0
```

将条目改为以下内容：

```
/dev/mapper/swapdev swap        swap    defaults        0 0
```

5. 使用多路径重建 initramfs 文件系统：

```
# dracut --force -H --add multipath
```

6. 关闭机器。

7. 引导机器。

8. 使其他路径对机器可见。

验证步骤

- 验证 swap 设备是否在多路径设备中：

```
# swapon -s
```

例如：

```
# swapon -s

Filename                Type        Size Used    Priority
/dev/dm-3               partition    4169724 0    -2
```

文件名应与多路径交换设备匹配。

```
# readlink -f /dev/mapper/swapdev
/dev/dm-3
```

7.4. 使用 DMSETUP 命令确定设备映射器条目

您可以使用 dmsetup 命令找出哪个设备映射器条目与多路径设备匹配。

流程

- 显示所有设备映射器设备及其主号码和副号码。副号码决定 dm 设备的名称。例如：副号码 3 与多路径设备 /dev/dm-3 对应。

```
# dmsetup ls
mpathd  (253:4)
mpathep1        (253:12)
mpathfp1        (253:11)
mpathb  (253:3)
mpathgp1        (253:14)
mpathhp1        (253:13)
mpatha  (253:2)
mpathh  (253:9)
mpathg  (253:8)
VolGroup00-LogVol01     (253:1)
mpathf  (253:7)
VolGroup00-LogVol00     (253:0)
mpathe  (253:6)
mpathbp1        (253:10)
mpathd  (253:5)
```

7.5. 管理 MULTIPATHD 守护进程

multipathd 命令可用于管理 multipathd 守护进程。

流程

- 查看 multipathd show maps 命令的输出的标准默认格式：

```
# multipathd show maps
name sysfs uuid
mpathc dm-0 360a98000324669436c2b45666c567942
```

- 有些 multipathd 命令包括 格式 选项，后跟通配符。使用以下命令显示可用通配符列表：

```
# multipathd show wildcards
```

- 以常规和原始格式显示多路径监控的多路径设备，使用带多路径通配符的格式字符串：

```
list|show maps|multipaths format $format
list|show maps|multipaths raw format $format
```

multipathd 命令支持以"raw"格式版本显示多路径设备和路径状态的格式命令。在原始格式中，不会打印标头，且不会添加字段来与标头匹配。相反，字段按照格式字符串中指定的内容完全相同。然后，此输出更易于用于脚本编写。您可以使用 multipathd show wildcard 命令显示格式字符串使用的通配符。

- 使用带有多路径通配符的格式字符串显示多路径监控的路径，使用常规和 raw 格式：

```
list|show paths format $format
list|show paths raw format $format
```

- 显示多路径显示的非原始格式和原始格式 之间的差别。请注意，采用 raw 格式的标头和列之间只能有一个空格：

```
# multipathd show maps format "%n %w %d %s"
name   uuid                             sysfs vend/prod/rev
mpathc 360a98000324669436c2b45666c567942 dm-0  NETAPP,LUN

# multipathd show maps raw format "%n %w %d %s"
mpathc 360a98000324669436c2b45666c567942 dm-0 NETAPP,LUN
```

其它资源

- multipathd(8)man page

第 8 章 删除存储设备

您可以从正在运行的系统中安全地删除存储设备，这有助于防止系统内存过载和数据丢失。

先决条件

- 在删除存储设备前，您必须确定您在 I/O 清除过程中因为系统内存负载增加而您有足够的可用内存。使用以下命令查看系统的当前内存负载和可用内存：

```
# vmstat 1 100
# free
```

- 红帽不推荐在以下系统中删除存储设备：
  - 空闲内存低于内存总量的 5%，每 100 个超过 10 个样本。
  - 交换是活跃的（非零 si，因此 vmstat 命令输出中的列）。

8.1. 安全删除存储设备

从正在运行的系统中安全地删除存储设备需要顶级的方法。从顶层（通常是应用程序或文件系统）开始，并在底层（即物理设备）上工作。

您可以通过多种方式使用存储设备，它们可以在物理设备之上有不同的虚拟配置。例如：您可以将设备的多个实例分组到多路径设备中，使其成为 RAID 的一部分，或者您可以将其成为 LVM 组的一部分。此外，设备可以通过文件系统访问，或者可以直接访问设备，如"原始"设备。

使用 top-to-bottom 方法时，您必须确保：

- 要删除的设备没有被使用
- 对该设备的所有待处理的 I/O 都会被清除

- 操作系统无法引用存储设备

8.2. 删除块设备

您可以从正在运行的系统中安全地删除块设备，以帮助防止系统内存过载和数据丢失。

警告

重新扫描 SCSI 总线或执行更改操作系统状态的其他操作，而无需遵循这个流程，因为 I/O 超时、设备被意外删除或数据丢失。

先决条件

- 如果您要删除多路径设备，且您无法访问其路径设备，请禁用多路径设备的队列：

```
# multipathd disablequeueing map multipath-device
```

这可让设备的 I/O 失败，允许使用该设备的应用程序关闭。

- 确定没有其他应用程序或服务正在使用您要删除的设备。
- 请确定从您要删除的设备备份数据。

流程

1. 使用 umount 命令卸载该设备上挂载的任何文件系统。
2. 从任何 MD RAID 阵列或者它所属的 LVM 卷中删除该设备。根据设备类型，执行以下步骤之一：

- 如果该设备是 LVM 组的成员，且它是一个多路径设备：

  a. 将数据移动到另一个设备中：

  ```
  # pvmove -b /dev/mapper/from-multipath-device /dev/mapper/to-multipath-device
  ```

  b. 从卷组中删除该设备：

  ```
  # vgreduce volume-group /dev/mapper/from-multipath-device
  ```

  c. 可选：从物理设备中删除 LVM 元数据：

  ```
  # pvremove /dev/mapper/from-multipath-device
  ```

- 如果您要删除多路径设备，请执行以下命令：

  a. 查看该设备的所有路径：

  ```
  # multipath -l
  ```

  稍后需要这个命令的输出。

  b. 清除 I/O 并删除多路径设备：

  ```
  # multipath -f multipath-device
  ```

- 如果该设备没有配置为多路径设备，或者设备配置为多路径设备，并且您之前将 I/O 传递给单个路径，请将任何未完成的 I/O 刷新到所有使用的设备路径：

  ```
  # blockdev --flushbufs device
  ```

  对于直接访问的设备非常重要，umount 或 vgreduce 命令不会清除 I/O。

- 如果您要删除 SCSI 设备，请执行以下命令：

  a. 删除对基于路径的设备名称的任何引用，如 */dev/sd*、*/dev/disk/by-path* 或 major:minor number（在系统上的应用程序、脚本或工具中）。这样可保证以后添加的不同设备不会为当前的设备错误。

  b. 从 SCSI 子系统中删除该设备的每个路径：

  ```
  # echo 1 > /sys/block/device-name/device/delete
  ```

  其中 *device-name* 从 multipath -l 命令的输出中检索（如果之前用作多路径设备）。

3. 从正在运行的系统中删除物理设备。请注意，当您删除此设备时，I/O 到其它设备不会停止。

其它资源

- multipath(8)、 pvmove(8)、 vgreduce(8)、 blockdev(8) 和 umount(8) man page。

第 9 章 DM 多路径故障排除

如果您在进行多路径配置时遇到问题，您可以检查这些问题。以下问题可能会导致多路径配置缓慢或无法正常工作：

多路径守护进程没有运行

如果您在进行多路径配置时遇到问题，请确保 multipathd 守护进程正在运行，如 设置 DM 多路径 中所述。multipathd 守护进程必须正在运行，才能使用多路径设备。

queue_if_no_path 功能的问题

如果使用 "1 queue_if_no_path" 选项配置多路径设备，那么在恢复一个或多个路径前，任何问题 I/O 的进程都会挂起。

9.1. 对 QUEUE_IF_NO_PATH 功能的问题进行故障排除

如果使用 "1 queue_if_no_path" 选项配置多路径设备，那么在恢复一个或多个路径前，任何问题 I/O 的进程都会挂起。要避免这种情况，请在 /etc/multipath.conf 文件中设置 no_path_retry *N* 参数，其中 N 是系统应该重试路径的次数。

如果您需要使用 "1 queue_if_no_path" 选项，并遇到这个问题，您可以在运行时为特定 LUN 禁用队列策略，以便所有路径都不可用。

流程

- 禁用特定设备的队列：

```
# multipathd disablequeueing map device
```

- 禁用所有设备的队列：

```
# multipathd disablequeueing maps
```

禁用某个设备队列后，它将保持禁用，直到 multipathd 重新启动或重新加载，或直至您执行以下命令之一：

- 将队列重置为特定设备的先前值：

```
# multipathd restorequeueing map device
```

- 将队列重置为所有设备的先前值：

```
# multipathd restorequeueing maps
```

9.2. 使用 MULTIPATHD 互动控制台进行故障排除

multipathd -k 命令是 multipathd 守护进程的互动接口。执行此命令将进入互动的多路径控制台。执行此命令后，您可以输入 help 来获取可用命令列表，Ctrl+D 退出。

使用 multipathd 互动控制台来排除您可能与您的系统相关的问题。

流程

- 在退出控制台前显示多路径配置，包括默认设置：

```
# multipathd -k
multipathd> show config
multipathd> Ctrl+D
```

- 确定多路径已获取对 multipath.conf 文件的任何更改：

```
# multipathd -k
multipathd> reconfigure
multipathd> Ctrl+D
```

- 确保路径检查程序正常工作：

```
# multipathd -k
multipathd> show paths
multipathd> Ctrl+D
```

- 您也可以直接从命令行运行单个 multipathd 交互式命令，而无需启动交互式控制台。例如，要检查多路径是否已获取对 multipath.conf 文件的任何更改，请运行：

```
# multipathd reconfigure
```

第 10 章 使用 EH_DEADLINE 配置存储错误恢复的最大时间

您可以配置最大允许的时间来恢复失败的 SCSI 设备。这个配置保证了 I/O 响应时间，即使存储硬件因为失败而变得无响应。

10.1. EH_DEADLINE 参数

SCSI 错误处理(EH)机制尝试在失败的 SCSI 设备上执行错误恢复。SCSI 主机对象 eh_deadline 参数允许您配置恢复的最大时间。配置的时间过期后，SCSI EH 会停止并重置整个主机总线适配器(HBA)。

使用 eh_deadline 可以缩短时间：

- 关闭失败的路径，
- 切换路径，或者
- 禁用 RAID 分片。

警告

当 eh_deadline 过期时，SCSI EH 会重置 HBA，这会影响那个 HBA 中的所有目标路径，而不仅仅是故障。如果由于其他原因无法使用冗余路径，则可能会出现 I/O 错误。仅在所有目标中有完全冗余的多路径配置时才启用 eh_deadline。

eh_deadline 参数的值以秒为单位指定。默认设置为 off，它会禁用时间限制并允许进行所有错误恢复。

eh_deadline 很有用的情况

在大多数情况下，您不需要启用 eh_deadline。在某些特定场景中，使用 eh_deadline 非常有用。例如，如果在光纤通道(FC)交换机和目标端口之间发生链接丢失，且 HBA 没有收到 Registered State Change Notifications(RSCN)。在这种情况下，I/O 请求和错误恢复命令会超时，而不是遇到错误。在这

个环境中设置 eh_deadline 会针对恢复时间设置上限。这可让失败的 I/O 在由 DM 多路径的另一个可用路径中检索。

在以下条件下，eh_deadline 参数不提供额外的好处，因为 I/O 和错误恢复命令会立即失败，这会导致 DM 多路径重试：

- 如果启用了 RSCN
- 如果 HBA 没有注册链接不可用

10.2. 设置 EH_DEADLINE 参数

这个过程配置 eh_deadline 参数的值来限制最大 SCSI 恢复时间。

流程

- 您可以使用以下方法之一配置 eh_deadline ：
  - multpath.conf 文件的 defaults 部分

    在 multpath.conf 文件的 defaults 部分，将 eh_deadline 参数设置为所需的秒数：

    ```
    # eh_deadline 300
    ```

    注意

    在 RHEL 8.4 中，使用 multpath.conf 文件的 defaults 部分设置 eh_deadline 参数是首选的方法。

    要使用此方法关闭 eh_deadline 参数，请将 eh_deadline 设置为 off。

  - sysfs

    将秒数写入 /sys/class/scsi_host/host<host-number>/eh_deadline 文件中。例如，

要在 SCSI 主机 6 上通过 sysfs 设置 eh_deadline 参数：

```
# echo 300 > /sys/class/scsi_host/host6/eh_deadline
```

要使用此方法关闭 eh_deadline 参数，请使用 echo off。

- 内核参数

  使用 scsi_mod.eh_deadline 内核参数为所有 SCSI HBA 设置默认值。

```
# echo 300 > /sys/module/scsi_mod/parameters/eh_deadline
```

要使用此方法关闭 eh_deadline 参数，请使用 echo -1。

其它资源

- 如何使用 udev 规则永久设置 eh_deadline 和 eh_timeout