Red Hat Enterprise Linux 9

RHEL 中逻辑卷的重复数据删除和压缩。

使用 VDO 来提高 LVM 存储容量

Last Updated: 2022-05-19

Red Hat Enterprise Linux 9 RHEL 中逻辑卷的重复数据删除和压缩。

使用 VDO 来提高 LVM 存储容量

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

本文档解释了如何在 LVM 中使用 Virtual Data Optimizer（VDO）功能来管理 RHEL 中对逻辑卷进行重复数据删除（deduplicate）和压缩。

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。详情请查看 CTO Chris Wright 的信息。

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，请创建一个新的问题单：
  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 LVM 的 VDO 介绍

Virtual Data Optimizer（VDO）为存储提供内联块级的重复数据删除（deduplication）、压缩和精简置备。您可以将 VDO 作为一个 LVM 逻辑卷类型（LV）来管理，类似于 LVM 精简置备的卷。

LVM（LVM-VDO）中的 VDO 卷由以下 LV 组成：

VDO 池 LV

这是用于 VDO LV 存储、重复数据删除和压缩的后端物理设备。VDO 池 LV 设置 VDO 卷的物理大小，即 VDO 可保存到磁盘中的数据量。
目前，每个 VDO 池 LV 只能有一个 VDO LV。因此，VDO 会单独压缩每个 VDO LV。换句话说，VDO 无法重复数据删除或压缩一些 VDO LV 共享的数据。

VDO LV

这是 VDO 池 LV 上的虚拟置备设备。VDO LV 设定 VDO 卷的置备和逻辑大小，即应用程序在重复数据删除和压缩发生前可写入卷的数据量。

表 1.1. LVM 和 LVM 精简置备的 VDO 组件的比较

	物理设备	置备的设备
LVM 上的 VDO	VDO 池 LV	VDO LV
LVM 精简配置	精简池（thin poll）	精简卷（thin volume）

由于 VDO 是迅速置备的，所以文件系统和应用程序只会看到使用中的逻辑空间，且不知道可用的实际物理空间。使用脚本来监控实际可用空间，并在使用超过阈值时生成警报：例如，当 VDO 池 LV 的使用超过 80%。

第 2 章 LVM-VDO 要求

LVM 上的 VDO 对其放置和系统资源有一定要求。

2.1. VDO 内存要求

每个 VDO 卷有不同的内存要求：

VDO 模块

VDO 需要固定的 38 MB RAM 和几个可变的数量：

- 每个 1 MB 的配置的块映射缓存需要 1.15 MB RAM。块映射缓存至少需要 150MB RAM。
- 每个 1 TB 逻辑空间需要 1.6 MB RAM。
- 由卷管理的每 1 TB 物理存储的 268 MB RAM。

UDS 索引

通用重复数据删除服务(UDS)至少需要 250 MB RAM，这也是重复数据删除使用的默认数量。您可以在格式化 VDO 卷时配置值，因为值还影响索引所需的存储量。
UDS 索引所需的内存由索引类型和重复数据删除窗口所需大小决定：

索引类型	重复数据删除窗口	备注
密度	每 1 GB RAM 为 1 TB	1GB 密度索引一般足以满足 4TB 物理存储空间。
稀疏	每 1 GB RAM 为 10 TB	1 GB 稀疏索引一般足以满足40TB 物理存储空间。

注意

使用默认设置的 2 GB slab 和 0.25dense 索引的 VDO 卷的最小磁盘用量需要大约 4.7 GB。这提供了在 0% 重复数据删除或压缩时写入的 2 GB 物理数据要少 2 GB。

这里的磁盘用量是默认 slab 大小和密度索引的总和。

UDS 稀疏索引功能是 VDO 推荐的模式。它依赖于数据的时序性，并尝试只保留内存中最相关的索引条目。使用稀疏索引，UDS 维护一个重复数据删除窗口，它是密度的10 倍，但使用相同数量的内存。

稀疏索引提供了最高的覆盖，但密度索引提供了更多的重复数据删除建议。对于大多数工作负载，如果内存量相同，则密度和稀疏索引间的重复数据删除率的不同会微不足道。

其他资源

- 按物理大小划分的 VDO 要求示例

2.2. VDO 存储空间要求

您可以将 VDO 卷配置为使用最多 256TB 物理存储。只有物理存储的某个部分可用来存储数据。本节提供了计算 VDO 管理的卷的可用空间大小的方法。

VDO 需要为两种类型的 VDO 元数据和 UDS 索引进行存储：

- 第一类 VDO 元数据对于每 4GB *物理贮存*使用 1 MB，再加上每个 slab 的额外的 1 MB。
- 第二类 VDO 元数据对于每 1GB *逻辑存储*使用 1.25 MB，并舍入到最近的 slab。
- UDS 索引所需的存储量取决于索引类型以及分配给索引的 RAM 量。对于每 1 GB RAM，密度 UDS 索引使用 17GB 存储，稀疏 UDS 索引使用 170 GB 存储。

其他资源

- 按物理大小划分的 VDO 要求示例
- VDO 中的 Lab 大小

2.3. 按物理大小划分的 VDO 要求示例

下表根据基础卷的物理大小提供 VDO 的最大系统要求。每个表都列出适合预期部署的要求，如主存储或备份存储。

具体数量取决于您的 VDO 卷的配置。

主存储部署

在主存储中，UDS 索引是物理大小的 0.01% 到 25%。

表 2.1. 主存储的存储和内存要求

物理大小	RAM 使用量：UDS	RAM 使用量：VDO	磁盘用量	索引类型
10GB–1TB	250MB	472MB	2.5GB	密度
2–10TB	1GB	3GB	10GB	密度
	250MB		22GB	稀疏
11–50TB	2GB	14GB	170GB	稀疏
51-100TB	3GB	27GB	255GB	稀疏
101–256TB	12GB	69GB	1020GB	稀疏

备份存储部署

在备份存储中，UDS 索引覆盖了备份组的大小，但小于物理大小。如果您预期备份集或物理大小在以后会增大，则需要把这个值加到索引大小中。

表 2.2. 备份存储的存储和内存要求

物理大小	RAM 使用量：UDS	RAM 使用量：VDO	磁盘用量	索引类型
10GB–1TB	250MB	472MB	2.5 GB	密度
2–10TB	2GB	3GB	170GB	稀疏
11–50TB	10GB	14GB	850GB	稀疏
51–100TB	20GB	27GB	1700GB	稀疏
101–256TB	26GB	69GB	3400GB	稀疏

2.4. 在存储堆栈中放置 LVM-VDO

您必须将特定的存储层放在 VDO 逻辑卷下，并在上面放置其他存储层。

您可以将thick 置备的层放在 VDO 的顶部，但您不能依赖该情况下精简置备的保证。因为 VDO 层是精简置备的，精简置备的效果适用于所有在它上面的层。如果您不监控 VDO 卷，您可能使用完在 VDO 以上的 thick-provisioned 卷的所有物理空间。

以下层支持的放置位于 VDO 下。不要将它们放在 VDO 上：

- DM Multipath
- DM Crypt
- Software RAID（LVM 或 MD RAID）

不支持以下配置：

- VDO 位于回送设备之上
- 加密的卷位于 VDO 之上
- VDO 卷中的分区
- 位于 VDO 卷之上的 RAID，比如 LVM RAID、MD RAID 或者其它类型
- 在 LVM-VDO 上部署 Ceph Storage

其他资源

- 堆栈 LVM 卷知识库文章

第 3 章 创建重复数据删除和压缩的逻辑卷

您可以创建使用 VDO 功能的 LVM 逻辑卷来重复数据删除和压缩数据。

3.1. LVM-VDO 部署情况

您可以以不同的方式部署 VDO on LVM（LVM-VDO）以提供重复数据删除的存储：

- 块访问
- 文件访问
- 本地存储
- 远程存储

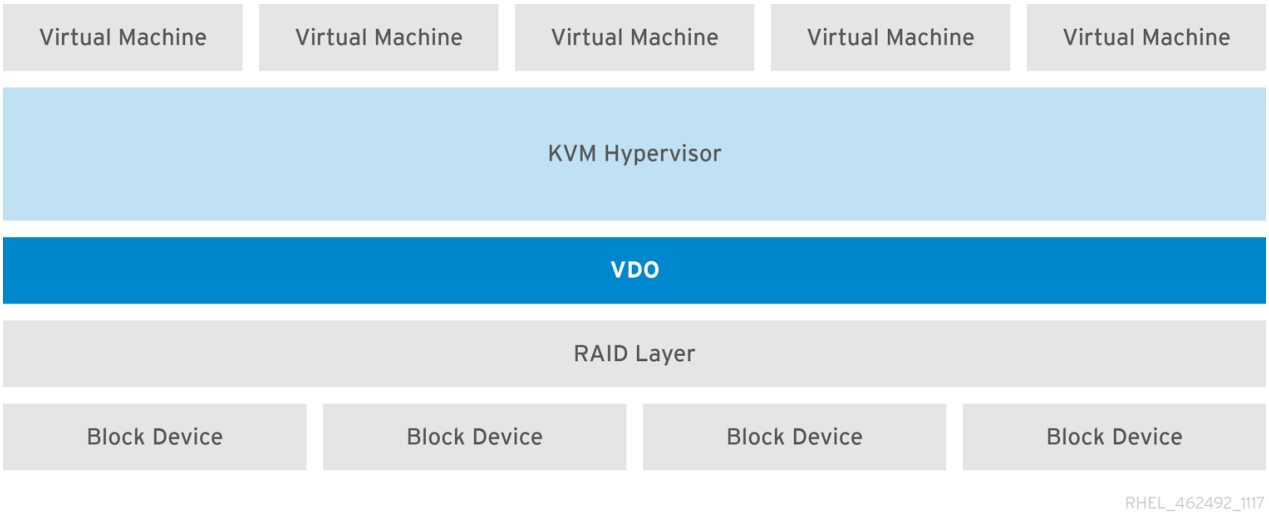
因为 LVM-VDO 会将重复数据删除存储作为常规逻辑卷（LV）形式公开，所以您可以在标准文件系统、iSCSI 和 FC 目标驱动程序或者统一存储中使用它。

注意

目前不支持在 LVM-VDO 上部署 Ceph Storage。

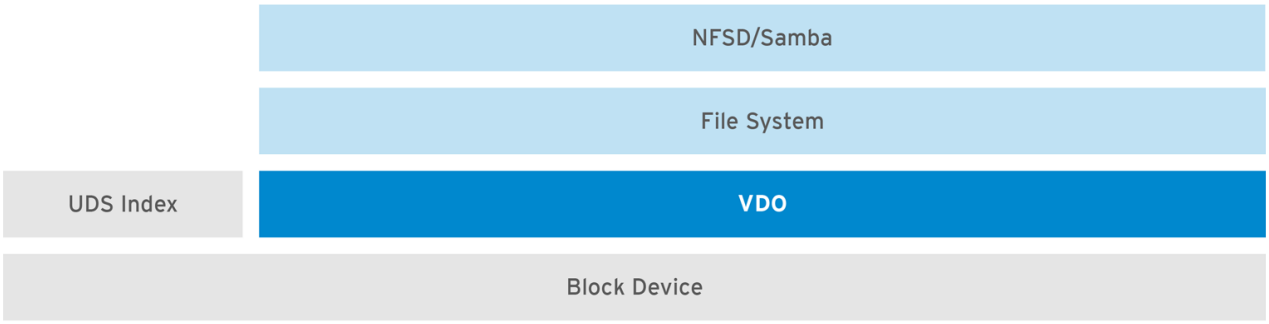
KVM

您可以在配置了直接附加存储的 KVM 服务器中部署 LVM-VDO。



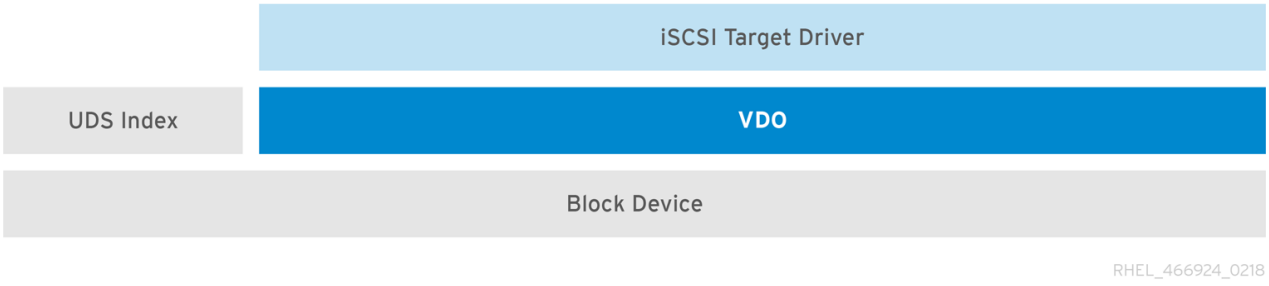
文件系统

您可以在 VDO LV 上创建文件系统，并将其公开给使用 NFS 服务器或 Samba 的 NFS 或 CIFS 用户。



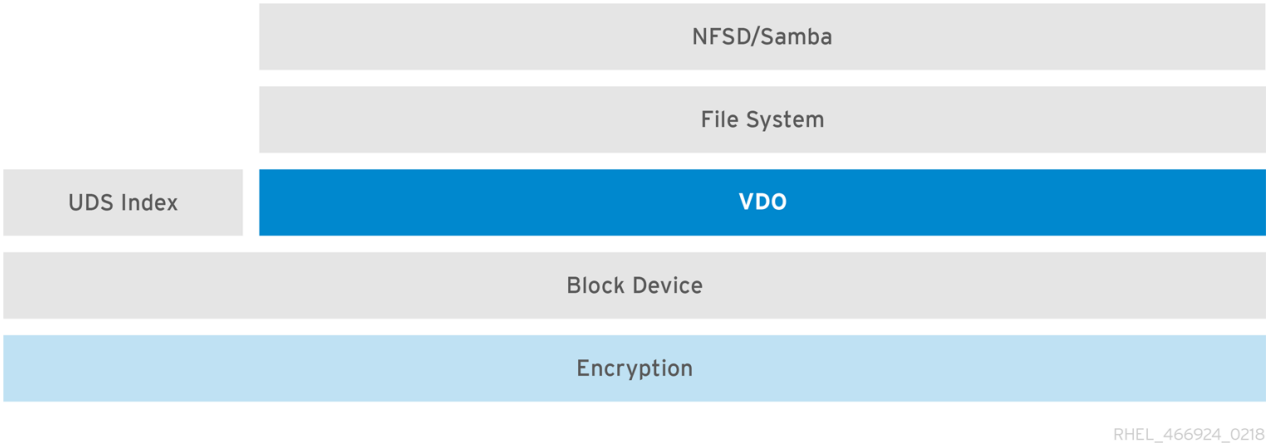
iSCSI 目标

您可以将 VDO LV 的整个导出为 iSCSI 目标到远程 iSCSI 启动器。



加密

DM Crypt 等设备映射器（DM）机制与 VDO 兼容。加密 VDO LV 卷有助于确保数据安全性，且所有 VDO LV 以上的文件系统仍会重复使用。



重要

如果对数据进行了重复数据删除，则在应用 VDO LV 上面应用加密层不会产生太大效果。在 VDO 可以对它们进行重复数据删除前，加密会使重复数据块不同。

始终将加密层放在 VDO LV 下。

3.2. LVM-VDO 卷的物理和逻辑大小

这部分论述了 VDO 可以使用的物理大小、可用物理大小和逻辑大小。

物理大小

这与分配给 VDO 池 LV 的物理区块大小相同。VDO 使用这个存储用于：

- 用户数据，这些数据可能会进行重复数据删除和压缩
- VDO 元数据，如 UDS 索引

可用物理大小

这是 VDO 可用于用户数据的物理大小的一部分。
它等同于物理大小减去元数据的大小，向下舍入到 slab 大小的倍数。

逻辑大小

这是 VDO LV 出现在应用程序中置备的大小。它通常大于可用的物理大小。VDO 目前支持任意逻辑卷大小最多为物理卷的 254 倍，但不能超过 4 PB。
当您设置 VDO 逻辑卷（LV）时，可以指定 VDO LV 出现的逻辑存储量。在托管活跃虚拟机或容器时，红帽建议使用 10:1 逻辑和物理比例置备存储：也就是说，如果您使用 1TB 物理存储，您将会把它显示为 10TB 逻辑存储。

如果没有指定 **--virtualsize** 选项，VDO 会将卷置备为 **1:1** 比例。例如，如果您将 VDO LV 放在 20GB VDO 池 LV 的上面，如果使用默认索引大小，VDO 为 UDS 索引保留 2.5 GB。剩余的 17.5 GB 为 VDO 元数据和用户数据提供。因此，要消耗的可用存储不超过 17.5 GB，且可能会因为组成实际 VDO 卷的元数据而减少。

其他资源

- 按物理大小划分的 VDO 要求示例

3.3. VDO 中的 LAB 大小

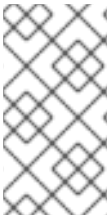
VDO 卷的物理存储被分成几个 slab。每个 slab 都是物理空间的连续区域。给定卷的所有 slab 的大小相同，可以是基于 128 MB 的 2 的指数的任何值，最大值为 32 GB。

默认的 slab 大小为 2 GB，用于在较小的测试系统中评估 VDO。单个 VDO 卷最多可有 8192 个 slabs。因此，在使用 2GB slab 的默认配置中，允许的最大物理存储为 16 TB。当使用 32GB 的 slab 时，允许的最大物理存储为 256 TB。VDO 总是保留至少一个整个 slab 来保存元数据，因此预留 slab 无法用于存储用户数据。

slab 大小不影响 VDO 卷的性能。

表 3.1. 根据物理卷大小推荐的 VDO slab 大小

物理卷大小	推荐的 slab 大小
10–99 GB	1 GB
100 GB – 1 TB	2 GB
2–256 TB	32 GB



注意

使用默认设置的 2 GB slab 和 0.25dense 索引的 VDO 卷的最小磁盘用量需要大约 4.7 GB。这提供了在 0% 重复数据删除或压缩时写入的 2 GB 物理数据要少 2 GB。

这里的磁盘用量是默认 slab 大小和密度索引的总和。

您可以通过向 **lvcreate** 命令提供 **--config 'allocation/vdo_slab_size_mb=*size-in-megabytes*'** 选项来控制 slab 大小。

3.4. 安装 VDO

此流程安装创建、挂载和管理 VDO 卷所需的软件。

步骤

- 安装 **vdo** 和 **kmod-kvdo** 软件包：

```
# dnf install vdo kmod-kvdo
```

3.5. 创建 LVM-VDO 卷

这个过程在 VDO 池 LV 中创建 VDO 逻辑卷（LV）。

先决条件

- 安装 VDO 软件。如需更多信息，请参阅安装 VDO。
- 在您的系统中有一个有可用存储容量的 LVM 卷组。

步骤

1. 为您的 VDO LV 选择一个名称，如 **vdo1**。您必须为系统中的每个 VDO LV 使用不同的名称和设备。
   在以下步骤中，将 *vdo-name* 替换为名称。
2. 创建 VDO LV：

```
# lvcreate --type vdo \
        --name vdo-name
        --size physical-size
        --virtualsize logical-size \
        vg-name
```

   - 使用您要放置 VDO LV 的现有 LVM 卷组的名称替换 *vg-name*。
   - 使用 VDO LV 存在的逻辑存储数量替换 *logical-size*。
   - 如果物理大小大于 16TiB，请添加以下选项以将卷的 slab 大小增加到 32GiB：

```
--config 'allocation/vdo_slab_size_mb=32768'
```

如果您在大于 16TiB 的物理大小中使用 2GiB 的默认 slab 大小，则 **lvcreate** 命令会失败并显示以下错误：

```
ERROR - vdoformat: formatVDO failed on '/dev/device': VDO Status: Exceeds maximum number of slabs supported
```

例 3.1. 为容器存储创建 VDO LV

例如，要为 1TB VDO 池中的容器存储创建 VDO LV，您可以使用：

```
# lvcreate --type vdo \
         --name vdo1
         --size 1T
         --virtualsize 10T \
         vg-name
```

重要

如果在创建 VDO 卷时发生故障，请删除要清理的卷。

3. 在 VDO LV 上创建文件系统：

- 对于 XFS 文件系统：

```
# mkfs.xfs -K /dev/vg-name/vdo-name
```

- 对于 ext4 文件系统：

```
# mkfs.ext4 -E nodiscard /dev/vg-name/vdo-name
```

其他资源

- **lvmvdo(7)** 手册页

3.6. 挂载 LVM-VDO 卷

这个过程会在 LVM-VDO 卷中手动挂载文件系统，也可以永久挂载文件系统。

先决条件

- 您的系统中有 LVM-VDO 卷。如需更多信息，请参阅创建 LVM-VDO 卷。

步骤

- 要手动将文件系统挂载到 LVM-VDO 卷中，请使用：

```
# mount /dev/vg-name/vdo-name mount-point
```

- 要将文件系统配置为在引导时自动挂载，请在 **/etc/fstab** 文件中添加行：
  - 对于 XFS 文件系统：

```
/dev/vg-name/vdo-name mount-point xfs defaults 0 0
```

- 对于 ext4 文件系统：

```
/dev/vg-name/vdo-name mount-point ext4 defaults 0 0
```

如果 LVM-VDO 卷位于需要网络的块设备中，如 iSCSI，请添加 **_netdev** 挂载选项。有关 **_netdev** 挂载选项的信息，请参阅 **systemd.mount(5)** 手册页。

其他资源

- **systemd.mount(5)** 手册页

3.7. 在 LVM-VDO 卷中更改压缩和重复数据删除设置

这个过程启用或禁用 VDO 池逻辑卷(LV)的压缩和重复数据删除。

注意

默认启用压缩和重复数据删除。

先决条件

- 您的系统中有 LVM-VDO 卷。

步骤

1. 要找出逻辑卷中是否启用或禁用压缩和重复数据删除：

```
# lvs -o+vdo_compression,vdo_deduplication
```

2. 查找运行活跃 VDOPoolLV 的重复数据删除索引的压缩状态和状态：

```
# lvs -o+vdo_compression_state,vdo_index_state
```

vdo_index_state 可以显示为 **error**, **close**, **opening**, **closing**, **online**, 和 **offline**。

3. 启用或禁用 VDOPoolLV 的压缩：

```
# lvchange --compression y|n  vg-name/vdopoolname
```

4. 为 VDOPoolLV 启用或禁用 deduplication：

```
# lvchange --deduplication y|n vg-name/vdopoolname
```

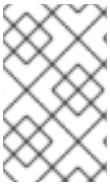
其他资源

- **lvmvdo(7)** 手册页

3.8. 使用虚拟数据优化器管理精简配置

可以通过配置精简配置的 VDO 卷准备以后扩展物理空间，以便解决 VDO 卷使用 100% 的条件。例如，在 **lvcreate** 操作中不使用 **-l 100%FREE** 而是使用例如 '95%FREE'，以确保稍后会根据需要进行恢复。这个步骤描述了如何解决这个问题：

- 卷耗尽空间
- 文件系统进入只读模式
- 卷报告的 ENOSPC



注意

解决 VDO 卷中高物理空间使用的最佳方法是删除未使用的文件，并使用在线丢弃这些未使用文件的块或 **fstrim** 程序。VDO 卷的物理空间只能增加到 8192 slab，对于一个默认 slab 大小为 2 GB 的 VDO 卷为 16 TB，或对于一个具有 32 GB 的 VDO 卷为 256 TB 。

在以下步骤中，将 *myvg* 和 *myvdo* 分别替换为卷组和逻辑卷名称。

先决条件

1. 已安装 **lvm2**、**vdo** d 和 **kmod-kvdo** 软件包。
2. 在您的系统中有一个有可用存储容量的 LVM 卷组。
3. 使用 **lvcreate --type vdo --name myvdo myvg -L logical-size-of-pool --virtualsize virtual-size-of-vdo** 命令的精简配置 VDO 卷。如需更多信息，请参阅 创建 LVM-VDO 卷。

步骤

1. 确定精简置备 VDO 卷的最佳逻辑大小

```
# vdostats myvg-vpool0-vpool

Device                 1K-blocks Used     Available  Use% Space saving%
myvg-vpool0-vpool   104856576  29664088 75192488   28%   69%
```

要计算空间节省率，请使用以下公式：

```
Savings ratio = 1 / (1 - Space saving%)
```

在本例中，

- 大约有 **3.22:1** 个空间节省率（大约 80 GB）。
- 如果对使用相同空间节省的数据写入 VDO 卷，则按比例增加数据集大小乘以 256 GB。
- 将这个数字调整到 200 GB 时，如果出现相同的空间节省率，则会产生一个具有安全可用磁盘空间的逻辑大小。

2. 监控 VDO 卷中的空闲物理空间：

```
# vdostats myvg-vpool0-vpool
```

可定期执行这个命令，以提供对 VDO 卷使用的和空闲物理空间的监控。

3. 可选：使用可用的 **/usr/share/doc/vdo/examples/monitor/monitor_check_vdostats_physicalSpace.pl** 脚本，查看 VDO 卷上的物理空间使用量警告：

```
# /usr/share/doc/vdo/examples/monitor/monitor_check_vdostats_physicalSpace.pl myvg-vpool0-vpool
```

4. 在创建 VDO 卷时，**dmeventd** 监控服务监控 VDO 卷中物理空间的使用情况。当 VDO 卷被创建或启动时，这会被默认启用。
在监控 VDO 卷时，使用 **journalctl** 命令查看日志中的 **dmeventd** 的输出：

```
lvm[8331]: Monitoring VDO pool myvg-vpool0-vpool.
...

lvm[8331]: WARNING: VDO pool myvg-vpool0-vpool is now 84.63% full.
lvm[8331]: WARNING: VDO pool myvg-vpool0-vpool is now 91.01% full.
lvm[8331]: WARNING: VDO pool myvg-vpool0-vpool is now 97.34% full.
```

5. 修复快要没有可用物理空间的 VDO 卷。当可以在 VDO 卷中添加物理空间时，但卷空间在可以增大前已满时，可能需要临时将 I/O 返回到卷。
要临时停止 I/O 到卷，请执行以下步骤，其中 VDO 卷 *myvdo* 包含挂载在 */users/homeDir* 路径中的文件系统：

   a. 冻结文件系统：

```
# xfs_freeze -f /users/homeDir

# vgextend myvg /dev/vdc2

# lvextend -l new_size myvg/vpool0-name

# xfs_freeze -u /users/homeDir
```

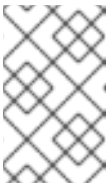
   b. 卸载文件系统：

```
# umount /users/homeDir

# vgextend myvg /dev/vdc2

# lvextend -l new_size myvg/vpool0-name

# mount -o discard /dev/myvg/myvdo /users/homeDir
```



注意

卸载或释放缓存数据的文件系统将产生缓存数据的写入，这可能会填满 VDO 卷的物理空间。当为 VDO 卷上的空闲物理空间设置监控阈值时，请考虑缓存的最大缓存文件系统数据量。

6. 可以使用 **fstrim** 程序清理文件系统不再使用的块。对 VDO 卷上的挂载的文件系统执行 **fstrim** 可能会导致该卷的可用空间增加。**fstrim** 工具将丢弃到 VDO 卷，然后用于删除对之前使用的块的引用。如果这些块中有单一引用，则使用物理空间。

   a. 检查 VDO stats 以查看当前可用空间量：

```
# vdostats --human-readable myvg-vpool0-vpool

 Device              Size  Used  Available Use%  Space saving%
myvg-vpool0-vpool  100.0G 95.0G 5.0G      95%   73%
```

b. 丢弃未使用块：

```
# fstrim /users/homeDir
```

c. 查看 VDO 卷的空闲物理空间：

```
# vdostats --human-readable myvg-vpool0-vpool


 Device              Size    Used   Available Use%  Space saving%
myvg-vpool0-vpool  100.0G   30.0G  70.0G     30%    43%
```

在这个示例中，在文件系统上执行 **fstrim** 后，丢弃可以返回 65G 物理空间以便在 VDO 卷中使用。

注意

丢弃较低级别的重复数据删除和压缩卷时，可能会回收物理空间，而不是丢弃更高水平的重复数据删除和压缩卷。具有高水平的重复数据删除和压缩卷可能需要进行更广泛的清理来回收物理空间，而不是只是丢弃尚未使用的块。

第 4 章 在 LVM-VDO 卷中修剪选项

您可以使用 **discard** 选项挂载文件系统，它会告知 VDO 卷未使用的空间。另一种选择是使用 **fstrim** 应用，它是一个按需丢弃的，或 **mount -o discard** 命令用于立即丢弃。

在使用 **fstrim** 应用时，管理员需要调度和监控额外的进程，而使用 **mount -o discard** 命令则可尽可能立即恢复空间。

请注意，目前建议使用 **fstrim** 应用程序丢弃未使用的块，而不是 **discard** 挂载选项，因为此选项的性能影响可能非常严重。因此，**nodiscard** 是默认值。

4.1. 在 VDO 中启用丢弃挂载选项

此流程在 VDO 卷中启用 **discard** 选项。

先决条件

- 您的系统中有 LVM-VDO 卷。

步骤

- 在卷中启用 **discard** ：

```
# mount -o discard /dev/vg-name/vdo-name mount-point
```

其他资源

- **XFS(5)**、**mount(8)** 和 **lvmvdo(7)** man page

4.2. 设置定期 TRIM 操作

这个过程在您的系统中启用调度的 TRIM 操作。

先决条件

- 您的系统中有 LVM-VDO 卷。

步骤

- 启用并启动计时器：

```
# systemctl enable --now fstrim.timer
```

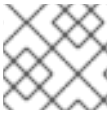
验证

- 验证计时器是否已启用：

```
# systemctl list-timers fstrim.timer
```

例 4.1. 验证过程的可能输出

```
# systemctl list-timers fstrim.timer
NEXT                         LEFT        LAST  PASSED  UNIT         ACTIVATES
Mon 2021-05-10 00:00:00 EDT  5 days left  n/a   n/a     fstrim.timer fstrim.service
```



注意

您没有对 VDO 卷的引用，因为 **fstrim.timer** 在所有挂载的文件系统中运行。

其他资源

- **fstrim(8)** man page