1. 安装LVM：

$sudo apt-get install lvm-common lvm2 evms

evms安装了才有什么vgdisplay、pvcreate等工具

1. Debian atitude下载lvm2 version of lvm2 2.02.95-8
2. 术语解释

　Logical Volumn Manager 简称LVM 是逻辑卷管理器

PV: Physical Volumn，物理卷，可以理解为磁盘分区

PE: Physical Extent, 每一个物理卷被划分为称为PE(Physical Extents)的基本单元，具有 唯一编号的PE是可以被LVM寻址的最小单元。PE的大小是可配置的，默认为4MB。

LV: Logical Volumn, 逻辑卷，在其上的文件系统看到的“磁盘分区”

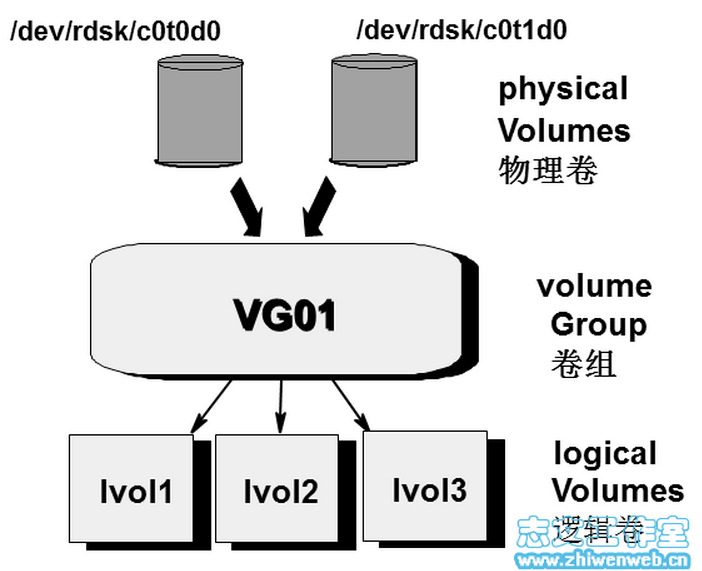
LE: Logical Extent, 逻辑卷也被划分为被称为LE(Logical Extents) 的可被寻址的基本单位。在同一个卷组中，LE的大小和PE是相同的，并且一一对应。

VG: Volumn Group，卷组，可以理解为存储池，它和上面几个东西的关系如下：

　　一个 PV 分成多个 PE，一个 LV 分成多个 LE；

　　一个 VG 由一个或多个 PV 组成；

一个 LE 从 VG 中分配出来，这个 LE 就是 mkfs 看到的分区，类似于通常情况下的物理分区 /dev/hda1 这样的东西。



1. 1）物理卷(Physical Volume)：物理卷就是指硬盘分区或从逻辑上与磁盘分区具有同样功能的设备(如RAID)，是LVM的基本存储逻辑块，但和基本的物理存储介质(如分区、磁盘等)比较，却包含有与LVM相关的管理参数。

2）卷组(Volume Group)：卷组由物理卷组成，类似于非LVM系统中的物理硬盘，可以在卷组上创建一个或多个"LVM分区"(逻辑卷)。

3）逻辑卷(Logical Volume)：LVM的逻辑卷类似于非LVM系统中的硬盘分区，在逻辑卷之上可以建立文件系统(比如/home或者/usr等)。

线性逻辑卷 (Linear Volumes)

一个线性逻辑卷聚合多个物理卷成为一个逻辑卷。比如,如果你有两个60GB硬盘，你可以生成120GB的逻辑卷。

4）条块化的逻辑卷(Striped Logical Volumes)：当你写数据到此逻辑卷中时,文件系统可以将数据放置到多个物理卷中。对于大量连接读写操作,它能改善数据I/O效率。

5）镜像的逻辑卷(Mirrored Logical Volumes)：镜像在不同的设备上保存一致的数据.数据同时被写入原设备及镜像设备.它提供设备之间的容错。

6）快照卷(Snapshot Volumes)：快照卷提供在特定瞬间的一个设备虚拟映像，当快照开始时，它复制一份对当前数据区域的改变，由于它优先执行这些改变，所以它可以重构当前设备的状态。

1. 物理卷命令：

#pvscan //在系统的所有磁盘中搜索已存在的物理卷

#pvdisplay 物理卷全路径名称 //用于显示指定物理卷的属性

#pvdata 物理卷全路径名称 //用于显示物理卷的卷组描述区域信息，用于调试目的。#pvchange –x|--allocation {y|n} 物理卷全路径名 //用于改变物理卷的分配许可设置

#pvcreate 设备全路径名 //用于在磁盘或磁盘分区上创建物理卷初始化信息，以便对该物理卷进行逻辑卷管理。

#pvmove 源物理卷全路径我[目的物理卷全路径名] //用于把某物理卷中的数据转移到同卷组中其他的特刊卷中。

1. 创建和管理LVM：
2. 创建分区：# fdisk -l /dev/sdb /dev/sdc

如果要将使用整块盘，可以通过下面的命令来覆盖磁盘上的原有分区信息：

#dd if=/dev/zero of=/dev/sdc bs=512 count=1

1. 创建物理卷：

# pvcreate /dev/sdb //将整个磁盘创建为物理卷

# pvcreate /dev/sdc1 //将单个分区创建为物理卷的命令为：

# pvcreate /dev/sdb /dev/sdc1 //同时生成多个卷

1. 扫描块设备：

# lvmdiskscan //通过lvmdiskscan可以看到那些设备成为了物理卷.

1. 显示物理卷： #pvs #pvscan #pvdisplay

//可以使用pvs,pvscan,pvdisplay来显示当前系统中的物理卷.

1. 移除物理卷： # pvremove /dev/sdc

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

1. 创建卷组：# vgcreate lvm\_test /dev/sdc1 /dev/sdb

命令第一个参数是指定该卷组的逻辑名：lvm\_test

1. 激活卷组：# vgchange -a y lvm\_test
2. 添加新的物理卷到卷组中：# fdisk -l /dev/sdc

# pvcreate /dev/sdc2 // 这里/dev/sdc2是新的物理卷

# vgextend lvm\_test /dev/sdc2

1. 显示卷组可以使用vgs和vgdisplay. # vgs # vgdisplay
2. 扫描磁盘生成缓存文件：# vgscan
3. 从卷组中删除一个物理卷：

要从一个卷组中删除一个物理卷，首先要确认要删除的物理卷没有被任何逻辑卷正在使用，就要使用pvdisplay命令察看一个该物理卷信息： # pvdisplay /dev/sdc2，显示： Allocated PE 0 (表示未被使用)

# vgreduce lvm\_test /dev/sdc2

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

1. 创建逻辑卷：# lvcreate -L 1500 –n test lvm\_test

该命令就在卷组lvm\_test上创建名字为test，大小为1500M的逻辑卷，并且设备入口为/dev/lvm\_test/test （lvm\_test为卷组名，test为逻辑卷名）。

如果希望创建一个使用全部卷组的逻辑卷，则需要首先察看该卷组的PE数，然后在创建逻辑卷时指定：

# vgdisplay lvm\_test| grep "Total PE"

Total PE 4731

# lvcreate -l 4731 lvm\_test -n test

1. 创建条块化的逻辑卷：# lvcreate -L 500M -i2 -n test lvm\_test

-i2指此逻辑卷在两个物理卷中条块化存放数据，默认一块大小为64KB.

1. 创建镜像的逻辑卷：

#lvcreate -L 52M -m1 -n test lvm\_test /dev/sdb1 /dev/sdc1 /dev/sdb2

-m1表示只生成一个单一镜像，镜像分别放在/dev/sdb1和/dev/sdc1上，镜像日志放在/dev/sdb2上。

1. 创建快照卷：

# lvcreate –size 10 –snapshot –name snaptest /dev/lvm\_test/test

1. 创建文件系统： 如使用ext3文件系统：# mkfs.ext3 /dev/lvm\_test/test

创建了文件系统以后，就可以加载并使用它：

# mkdir /mnt/test

# mount /dev/lvm\_test/test /mnt/test

如果希望系统启动时自动加载文件系统，则还需要在/etc/fstab中添加内容：

/dev/lvm\_test/test /mnt/test ext3 defaults 1 2

1. 删除一个逻辑卷：

删除逻辑卷以前首先需要将其卸载，然后删除：

# umount /dev/lvm\_test/test

# lvremove /dev/lvm\_test/test

1. 扩展逻辑卷大小：

# lvextend -L12G /dev/lvm\_test/test

上面的命令就实现将逻辑卷test的大小扩大为12G。

# lvextend -L+1G /dev/lvm\_test/test

上面的命令就实现将逻辑卷test的大小增加1G。

增加了逻辑卷的容量以后，就需要修改文件系统大小以实现利用扩充的空间。对于希望调整被加载的文件系统大小,使用ext2online(ext2resize) 或 resize2fs。

#df -h

# ext2online /dev/lvm\_test/test

一般建议最好将文件系统卸载，调整大小，然后再加载：

# umount /dev/lvm\_test/test

#resize2fs /dev/lvm\_test/test

# mount /dev/lvm\_test/test /mnt/test

1. 减少逻辑卷大小：

使用lvreduce即可实现对逻辑卷的容量，同样需要首先将文件系统卸载：

# umount /mnt/test

# e2fsck -f /dev/lvm\_test/test

# resize2fs /dev/lvm\_test/test 11G

# lvreduce -L -1.992G /dev/lvm\_test/test （少2个PE的大小）

# resize2fs /dev/lvm\_test/test

// (通过此命令确认是否文件系统大小与收缩后的逻辑卷大小匹配）

# mount /dev/lvm\_test/test /mnt/test

1. 显示逻辑卷：

通过lvscan,lvs及lvdisplay可以察看当前系统中存在的逻辑卷。

# lvdisplay

#lvscan

#lvs -o +devices

1. 使用过滤控制LVM的设备扫描

通过编辑/etc/lvm/lvm.conf 中的filter段，来定义过滤哪些设备要扫描。

filter =[ "a|/dev/sd.\*|", "a|/dev/hd.\*|", "r|.\*|" ]

1. 在线数据迁移：

通过pvmove可以将一个PV上的数据迁移到新的PV上，也可以将PV上的某个LV迁移到另一个PV上。

# lvs -o +devices

# pvmove -n test /dev/sdb1 /dev/sdc1

# lvs -o +devices

1. 删除卷组

按照顺序卸载文件系统，删除逻辑卷，然后删除卷组。

#umount /mnt/test

# lvremove /dev/lvm\_test/test

1. 故障排查：

通过在lvs,vgs后加-P可以更好的查看失败设备。

#vgs -a -o +devices –P

#lvs -a -o +devices -P

1. 物理卷命令：

pvcreate 创建LVM物理卷

pvdisplay 显示已创建的物理卷信息

pvchange 设置PV的性能，允许或拒绝从这个磁盘上分配置另外的PE。

pvmove 在一个卷组中从源到目的移动已经分配的PE

pvremove 删除一个物理卷(也就是从物理卷中删除一个LVM标签)

1. 卷组命令：

vgcreate 创建卷组

vgdisplay 显示卷组的信息

vgchange 激活或者撤消卷组，允许卷组使用或不使用quorum来挂载

vgchange -a y 激活所有卷组 (开机执行,redhat可在/etc/rc.d/rc.sysinit系统启动初始化脚本里可以找到)

vgchange -a n 关闭所有卷组(提示:必须在umount所有的文件系统后,才能成功执行

vgextend 通过添加磁盘扩充卷组

vgreduce 删除物理卷缩减卷组，vgscan 扫描所有磁盘寻找卷组

vgsync 同步镜像

remove 删除卷组

vgexport 从系统中删除一个卷组，但不修改在基于物理卷的信息

vgimport 通过扫描使用vgexport命令输出的物理卷在系统中增加一个

Vgcfgbackup 保存卷组的配置信息，记住一个卷组有一个或多个物理卷组成

vgcfgrestore 恢复卷的配置信息

vgscan 读取系统中创建的所有卷组

1. 逻辑卷命令：

lvcreate 生成逻辑卷

lvdisplay 显示逻辑卷的信息

lvchange 改变逻辑卷的特性，包括可用性、调度策略、权限、块重定位、分配策略、镜像缓存的可用性lvextend 增加逻辑卷的空间

extendfs 扩展文件系统的大小

lvreduce 减少逻辑卷的空间

lvremove 删除逻辑卷

lvsplit 分割镱像的逻辑卷

lvmerge 合并lvsplit过的逻辑卷

lvsync 同步逻辑卷

lvmmigrate 在一个分区中为逻辑卷迁移准备一个ROOT文件

lvlnboot 用来建立root,primary swap ,or dump逻辑卷

lvlnboot 删除lvlnboot建立的逻辑卷

1. 其他命令：

fdisk -l :查看系统中都认到了那些物理硬盘

mkfs:基于逻辑卷创建一个相应类型的文件系统

mkdir -p $mount\_piont:创建一个挂载目录

创建好的文件系统位于:

/dev/$create\_vg\_name/$lv\_name

mount /dev/$create\_vg\_name/$lv\_name $mount\_piont:挂载文件系统

1. 卷组命令：

#vgscan //检测系统中所有磁盘

#vgck [卷组名] //用于检查卷组中卷组描述区域信息的一致性。

#vgdisplay [卷组名] //显示卷组的属性信息

#vgrename 原卷组名 新卷组名

#vgchange -a y|n [卷组名] //改变卷组的相应属性。是否可分配

#vgchange -l 最大逻辑卷数 //卷组可容纳最大逻辑卷数

#vgchange -x y|n [卷组名] //卷是否有效

#vgmknodes [卷组名|卷组路径] //用于建立（重新建立）已有卷组目录和其中的设备文件卷组配置的备份与恢复命令

#vgcfgbackup [卷组名] //把卷组中的VGDA信息备份到“/etc/lvmconf”目录中的文件

#vgcfgrestore -n 卷组名 物理卷全路命名 //从备份文件中必得指定物理卷的信息卷组的建立与删除命令

#vgcreate 卷组名 物理卷全路径名[物理卷全路径名]

#vgmove 卷组名

卷组的扩充与缩小命令：

#vgextend 卷组名 物理卷全路径名[物理卷全路径名]

#vgreduce 卷组名 物理卷全路径名[物理卷全路径名]

卷组的合并与拆分：

#vgmerge 目的卷组名 源卷组名 //合并两个已经存在的卷组，要求两个卷组的物理区域大小相等且源卷组是非活动的。

#vgsplit 现有卷组 新卷组 物理卷全路径名[物理卷全路径名]

卷组的输入与输出命令：

#vgexport 卷组名

#vgimport 卷组名 卷组中的物理卷[卷组中的物理卷]

1. 逻辑卷命令

一般命令

#lvscan#lvdisplay 逻辑卷全路径名[逻辑卷全路径名]

#lvrename 旧逻辑卷全路径名 新逻辑卷全路径名

#lvrename 卷组名 旧逻辑卷名 新逻辑卷名

#lvchange#e2fsadm -L +|- 逻辑卷增减量 逻辑卷全路径名

逻辑卷的创建与删除命令

#lvcreate

#lvremove

逻辑卷的扩充与缩小命令

#lvextend -L|--size +逻辑卷大小增量 逻辑卷全路径名

#lvreduce q -L|--size +逻辑卷减小量 逻辑卷全路径名

1. 其它事项：