lvm

lvm（LogicalVolumeManager）逻辑盘卷管理

# 前言

每个Linux使用者在安装Linux时都会遇到这样的困境：在为系统分区时，如何精确评

估和分配各个硬盘分区的容量，因为系统管理员不但要考虑到当前某个分区需要的容量，还要预见该分区以后可能需要的容量的最大值。因为如果估 计不准确，当遇到某个分区不够用时管理员可能甚至要备份整个系统、清除硬盘、重新对硬盘分区，然后恢复数据到新分区。

虽然有很多动态调整磁盘的工具可以使用，例如PartitionMagic等等，但是它并不能完全解决问题，因为某个分区可能会再次被耗尽；另外一个方面这需要 重新引导系统才能实现，对于很多关键的服务器，停机是不可接受的，而且对于添加新硬盘，希望一个能跨越多个硬盘驱动器的文件系统时，分区调整程序就不能解 决问题。

因此完美的解决方法应该是在零停机前提下可以自如对文件系统的大小进行调整，可以方便实现文件系统跨越不同磁盘和分区。幸运的是Linux提供的逻辑盘卷管理（LVM，LogicalVolumeManager）机制就是一个完美的解决方案。

LVM是逻辑盘卷管理（LogicalVolumeManager）的简称，它是Linux环境下对磁盘分区进行管理的一种机制，LVM是建立在硬盘和 分区之上的一个逻辑层，来提高磁盘分区管理的灵活性。通过LVM系统管理员可以轻松管理磁盘分区，如：将若干个磁盘分区连接为一个整块的卷组 （volumegroup），形成一个存储池。管理员可以在卷组上随意创建逻辑卷组（logicalvolumes），并进一步在逻辑卷组上创建文件系 统。管理员通过LVM可以方便的调整存储卷组的大小，并且可以对磁盘存储按照组的方式进行命名、管理和分配，例如按照使用用途进行定义：“development”和“sales”，而不是使用物理磁盘名“sda”和“sdb”。而且当系统添加了新的磁盘，通过LVM管理员就不必将磁盘的 文件移动到新的磁盘上以充分利用新的存储空间，而是直接扩展文件系统跨越磁盘即可。

# 基本术语

前面谈到，LVM是在磁盘分区和文件系统之间添加的一个逻辑层，来为文件系统屏蔽下层磁盘分区布局，提供一个抽象的存储卷，在存储卷上建立文件系统。首先我们讨论以下几个LVM术语：

\*物理存储介质（PhysicalStorageMedia）

指系统的物理存储设备：磁盘，如：/dev/hda、/dev/sda等，是存储系统最底层的存储单元。

\*物理卷（Physical Volume，PV）

指磁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。

\*卷组（Volume Group，VG）

类似于非LVM系统中的物理磁盘，其由一个或多个物理卷PV组成。可以在卷组上创建一个或多个LV（逻辑卷）。

\*逻辑卷（Logical Volume，LV）

类似于非LVM系统中的磁盘分区，逻辑卷建立在卷组VG之上。在逻辑卷LV之上可以建立文件系统（比如/home或者/usr等）。

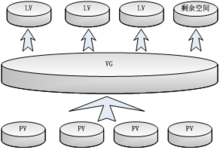
\*物理块（Physical Extent，PE）

每一个物理卷PV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。所以物理卷（PV）由大小等同的基本单元PE组成。

\*逻辑块（Logical Extent，LE）

逻辑卷LV也被划分为可被寻址的基本单位，称为LE。在同一个卷组中，LE的大小和PE是相同的，并且一一对应。

图所示LVM抽象模型，展示了PV、VG、LV三者之间关系：



和非LVM系统将包含分区信息的元数据保存在位于分区的起始位置的分区表中一样，逻辑卷以及卷组相关的元数据也是保存在位于物理卷起始处的VGDA（卷组描述符区域）中。VGDA包括以下内容：PV描述符、VG描述符、LV描述符、和一些PE描述符。

系统启动LVM时激活VG，并将VGDA加载至内存，来识别LV的实际物理存储位置。当系统进行I/O操作时，就会根据VGDA建立的映射机制来访问实际的物理位置。

# 创建管理

要创建一个LVM系统，一般需要经过以下步骤：

## 1、创建分区

使用分区工具（如：fdisk等）创建LVM分区，方法和创建其他一般分区的方式是一样的，区别仅仅是LVM的分区类型为8e。

## 2、创建物理卷

创建物理卷的命令为pvcreate，利用该命令将希望添加到卷组的所有分区或者磁盘创建为物理卷。将整个磁盘创建为物理卷的命令为：

#pvcreate /dev/hdb

将单个分区创建为物理卷的命令为：

#pvcreate /dev/hda5

## 3、创建卷组

创建卷组的命令为vgcreate，将使用pvcreate建立的物理卷创建为一个完整的卷组：

#vgcreate web\_document /dev/hda5 /dev/hdb

vgcreate命令第一个参数是指定该卷组的逻辑名：web\_document。后面参数是指定希望添加到该卷组的所有分区和磁盘。vgcreate 在创建卷组web\_document以外，还设置使用大小为4MB的PE（默认为4MB），这表示卷组上创建的所有逻辑卷都以4MB为增量单位来进行扩充 或缩减。由于内核原因，PE大小决定了逻辑卷的最大大小，4MB的PE决定了单个逻辑卷最大容量为256GB，若希望使用大于256G的逻辑卷则创建卷组时指定更大的PE。PE大小范围为8KB到512MB，并且必须总是2的倍数（使用-s指定，具体请参考manvgcreate）。（centos 6.2系统已发现没有这种限制）

## 4、激活卷组

为了立即使用卷组而不是重新启动系统，可以使用vgchange来激活卷组：

#vgchange -ay web\_document

## 5、添加新的物理卷到卷组中

当系统安装了新的磁盘并创建了新的物理卷，而要将其添加到已有卷组时，就需要使用vgextend命令：

#vgextend web\_document /dev/hdc1

这里/dev/hdc1是新的物理卷。

## 6、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：

如果某个物理卷正在被逻辑卷所使用，就需要将该物理卷的数据备份到其他地方，然后再删除。删除物理卷的命令为vgreduce：

#vgreduce web\_document /dev/hda1

## 7、创建逻辑卷

创建逻辑卷的为命令lvcreate：

#lvcreate -L1500 -n www1 web\_document

该命令就在卷组web\_document上创建名字为www1，大小为1500M的逻辑卷，并且设备入口为 /dev/web\_document/www1（web\_document为卷组名，www1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 web\_document | grep"TotalPE"

TotalPE45230

#lvcreate -l45230 web\_document -n www1

## 8、创建文件系统

笔者推荐使用reiserfs文件系统，来替代ext2和ext3：

创建了文件系统以后，就可以加载并使用它：

#mkdir/data/wwwroot

#mount /dev/web\_document/www1/data/wwwroot

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/web\_document/www1/data/wwwrootreiserfsdefaults12

## 9、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/web\_document/www1

#lvremove /dev/web\_document/www1

lvremove--doyoureallywanttoremove"/dev/web\_document/www1"?[y/n]:y

lvremove--doingautomaticbackupofvolumegroup"web\_document"

lvremove--logicalvolume"/dev/web\_document/www1"successfullyremoved

## 10、扩展逻辑卷大小

LVM提供了方便调整逻辑卷大小的能力，扩展逻辑卷大小的命令是lvextend：

#lvextend -L12G /dev/web\_document/www1

lvextend--extendinglogicalvolume"/dev/web\_document/www1"to12GB

lvextend--doingautomaticbackupofvolumegroup"web\_document"

lvextend--logicalvolume"/dev/web\_document/www1"successfullyextended

上面的命令就实现将逻辑卷www1的大小扩招为12G。

#lvextend -L +1G /dev/web\_document/www1

lvextend--extendinglogicalvolume"/dev/web\_document/www1"to13GB

lvextend--doingautomaticbackupofvolumegroup"web\_document"

lvextend--logicalvolume"/dev/web\_document/www1"successfullyextended

上面的命令就实现将逻辑卷www1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。笔者推荐使用reiserfs文件系统来替代ext2或者ext3。因此这里仅 仅讨论reiserfs的情况。Reiserfs文件工具提供了文件系统大小调整工具：resize\_reiserfs。对于希望调整被加载的文件系统大 小：

#resize\_reiserfs -f /dev/web\_document/www1

一般建议最好将文件系统卸载，调整大小，然后再加载：

#umount /dev/web\_document/www1

#resize\_reiserfs /dev/web\_document/www1

#mount-treiserfs /dev/web\_document/www1/data/wwwroot

对于使用ext2或ext3文件系统的用户可以考虑使用工具

ext2resize。

## 11、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 /data/wwwroot

#resize\_reiserfs -s -2G /dev/web\_document/www1

#lvreduce -L -2G /dev/web\_document/www1

#mount-treiserfs /dev/web\_document/www1/data/wwwroot

# 优点

LVM通常用于装备大量磁盘的系统，但它同样适于仅有一、两块硬盘的小系统。

小系统使用LVM的益处

传统的文件系统是基于分区的，一个文件系统对应一个分区。这种方式比较直观，但不易改变：

1. 不同的分区相对独立，无相互联系，各分区空间很易利用不平衡，空间不能充分利用；
2. 当一个文件系统/分区已满时，无法对其扩充，只能采用重新分区/建立文件系统，非常麻烦；或把分区中的数据移到另一个更大的分区中；或采用符号连接的方式使用其它分区的空间。

3.如果要把硬盘上的多个分区合并在一起使用，只能

采用再分区的方式，这个过程需要数据的备份与恢复。当采用LVM时，情况有所不同：

3.1.硬盘的多个分区由LVM统一为卷组管理，可以方便的加入或移走分区以扩大或减小卷组的可用容量，充分利用硬盘空间；

3.2.文件系统建立在逻辑卷上，而逻辑卷可根据需要改变大小（在卷组容量范围内）以满足要求；

3.3.文件系统建立在LVM上，可以跨分区，方便使用；

大系统使用LVM的益处

在使用很多硬盘的大系统中，使用LVM主要是方便管理、增加了系统的扩展性。

在一个有很多不同容量硬盘的大型系统中，对不同的用户的空间分配是一个技巧性的工作，要在用户需求与实际可用空间中寻求平衡。

用户/用户组的空间建立在LVM上，可以随时按要求增大，或根据使用情况对各逻辑卷进行调整。当系统空间不足而加入新的硬盘时，不必把用户的数据从原硬盘迁移到新硬盘，而只须把新的分区加入卷组并扩充逻辑卷即可。同样，使用LVM可以在不停服务的情况下。把用户数据从旧硬盘转移到新硬盘空间中去。

# 实现

Linux中实现LVM的方法有两种：一种是在安装Linux时利用Disk Druid程序在图形化界面下实现；另一种是利用LVM命令在字符界面下实现，下面的过程是基于后一种方法实现的。

实现环境如下：操作系统为Red Hat Enterprise LinuxAS 4 ，默认安装的LVM管理工具包为LVM2 2.00.31 ，硬盘为36GBSCSI接口共计2块，其中sda存放系统文件及数据，sdb、sdc用于实现LVM。

## 1．准备物理分区（Physical Partions）

首先，我们需要选择用于 LVM 的物理存储器。这些通常是标准分区，但也可以是已创建的 Linux Software RAID 卷。这里我利用fdisk命令，将sdb、sdc两块磁盘分了两个区sdb1、sdc1， 通过fdisk的t指令指定分区为8e类型（Linux LVM）。如图1所示。

图1 准备物理分区的指令执行界面

## 2、创建物理卷PV（Physical Volumes）

物理卷（Physical Volumes）简称PV，是在磁盘的物理分区或与磁盘分区具有同样功能的设备（如RAID）上创建而来。它只是在物理分区中划出了一个特殊的区域，用于记载与LVM相关的管理参数。

创建物理卷的命令是 pvcreate :

[root@li2 ~]# pvcreate /dev/sdb1

Physical volume "/dev/sdb1" successfully created

[root@li2 ~]# pvcreate /dev/sdc1

Physical volume "/dev/sdc1" successfully created

以上命令分别将/dev/sdc1、/dev/sdb1初始化成物理卷，使用物理卷显示命令pvdisplay查看物理卷情况如下：

[root@li2 ~]# pvdisplay

--- NEW Physical volume ---

PV Name /dev/sdb1

VG Name

PV Size 36.00 GB

Allocatable NO

PE Size (KByte) 0

Total PE 0

Free PE 0

Allocated PE 0

PV UUID QDmnUd-tuvH-U4Hn-n5Ry-zGRT-O1yK-67Dxbb

--- NEW Physical volume ---

PV Name /dev/sdc1

VG Name

PV Size 36.00 GB

Allocatable NO

PE Size (KByte) 0

Total PE 0

Free PE 0

Allocated PE 0

PV UUID NDBf68-6qrD-9hE6-Rotv-RdxL-Azvv-7NlC0S

## 3．创建卷组VG（Volume Groups）

卷组（Volume Group）简称VG，它是一个或者多个物理卷的组合。卷组将多个物理卷组合在一起，形成一个可管理的单元，它类似于非LVM系统中的物理硬盘。

创建卷组的命令为vgcreate，下面利用它创建了一个名为“lvmdisk”的卷组，该卷组包含/dev/sdb1、/dev/sdc1两个物理卷。

[root@li2 ~]# vgcreate lvmdisk /dev/sdb1 /dev/sdc1

Volume group "lvmdisk" successfully created

使用卷组查看命令vgdisplay显示卷组情况：

[root@li2 ~]# vgdisplay

--- Volume group ---

VG Name lvmdisk

System ID

Format lvm2

Metadata Areas 2

Metadata Sequence No 1

VG Access read/write

VG Status resizable

MAX LV 0

Cur LV 0

Open LV 0

Max PV 0

Cur PV 2

Act PV 2

VG Size 71.98 GB

PE Size 4.00 MB

Total PE 18428

Alloc PE / Size 0 / 0

Free PE / Size 18428 / 71.98 GB

VG UUID SARfuj-wAUI-od81-VWAc-A1nt-aaFN-JWaPVf

当多个物理卷组合成一个卷组后时，LVM会在所有的物理卷上做类似格式化的工作，将每个物理卷切成一块一块的空间，这一块一块的空间就称为PE（Physical Extent ），它的默认大小是4MB。

由于受内核限制的原因，一个逻辑卷（Logic Volume）最多只能包含65536个PE（Physical Extent），所以一个PE的大小就决定了逻辑卷的最大容量，4 MB 的PE决定了单个逻辑卷最大容量为 256 GB，若希望使用大于256G的逻辑卷，则创建卷组时需要指定更大的PE。在Red Hat Enterprise Linux AS 4中PE大小范围为8 KB 到 16GB，并且必须总是 2 的倍数。

例如，如果希望使用 64 MB 的PE创建卷组，这样逻辑卷最大容量就可以为4 TB，命令如下：

# vgcreate － 64MB lvmdisk /dev/sdb1 /dev/sdc1

## 4．创建逻辑卷LV（Logical Volumes）

逻辑卷（Logical Volumes）简称LV，是在卷组中划分的一个逻辑区域，类似于非LVM系统中的硬盘分区。

创建逻辑卷的命令为lvcreate，通过下面的命令，我们在卷组lvmdisk上创建了一个名字为pldy1的逻辑卷，大小为15GB，其设备入口为/dev/lvmdisk/pldy1。

[root@li2 dev]# lvcreate -L 15G -n pldy1 lvmdisk

Logical volume "pldy1" created

也可以使用-l参数，通过指定PE数来设定逻辑分区大小。

例如，希望创建一个使用全部空间的逻辑卷，需要先查清卷组中的PE总数，通过上面的vgdisplay命令查得当前卷组PE总数为18428，命令如下：

# lvcreate -l 18428 -n pldy1 lvmdisk

当逻辑卷创建成功后，可以使用lvdisplay命令查看逻辑卷情况：

[root@li2 ~]# lvdisplay

--- Logical volume ---

LV Name /dev/lvmdisk/pldy1

VG Name lvmdisk

LV UUID FQcnm3-BMyq-NkJz-hykw-9xg1-Qy8d-8UeGCN

LV Write Access read/write

LV Status available

# open 0

LV Size 15.00 GB

Current LE 3840

Segments 1

Allocation inherit

Read ahead sectors 0

Block device 253:0

同卷组一样，逻辑卷在创建的过程中也被分成了一块一块的空间，这些空间称为LE（Logical Extents），在同一个卷组中，LE的大小和PE是相同的，并且一一对应。

## 5．创建文件系统

在逻辑卷上创建ext3文件系统：

[root@li2 ~]# mkfs -t ext3 /dev/lvmdisk/pldy1

创建了文件系统以后，就可以加载并使用了：

[root@li2 ~]# mkdir /opt/Oracle

[root@li2 ~]# mount /dev/lvmdisk/pldy1 /opt/Oracle

为了在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/lvmdisk/pldy1 /opt/Oracle ext3 defaults 1 2

管理LVM

LVM的最大好处就是可以动态地调整分区大小，而无须重新启动机器，下面让我们来体验一下吧！继续上面的实例，现假设逻辑卷/dev/lvmdisk/pldy1空间不足，需要增加其大小，

## 6、我们分两种情况讨论：

### 1．卷组中有剩余的空间

通过vgdisplay命令可以检查当前卷组空间使用情况：

[root@li2 ~]# vgdisplay

--- Volume group ---

VG Name lvmdisk

System ID

Format lvm2

Metadata Areas 2

Metadata Sequence No 2

VG Access read/write

VG Status resizable

MAX LV 0

Cur LV 1

Open LV 0

Max PV 0

Cur PV 2

Act PV 2

VG Size 71.98 GB

PE Size 4.00 MB

Total PE 18428

Alloc PE / Size 3840 / 15.00 GB

Free PE / Size 14588 / 56.98 GB

VG UUID SARfuj-wAUI-od81-VWAc-A1nt-aaFN-JWaPVf

确定当前卷组剩余空间56.98GB，剩余PE数量为14588个。在这里将所有的剩余空间全部增加给逻辑卷 /dev/lvmdisk/pldy1。

[root@li2 Oracle]# lvextend -l+14588 /dev/lvmdisk/pldy1

Extending logical volume pldy1 to 56.98 GB

Logical volume pldy1 successfully resized

上面的命令使用了-l+14588参数，它的意思是给指定的逻辑卷增加14588个PE。如果不是将全部空间都使用，还可使用其他形式的lvextend命令。

例如将逻辑卷/dev/lvmdisk/pldy1增加5GB的空间，使其空间达到20GB，可写成： “# lvextend -L+5G /dev/lvmdisk/pldy1”或“# lvextend －L20G /dev/lvmdisk/pldy1”。增加了逻辑卷容量后，就要通过ext2online命令修改文件系统的大小了。

[root@li2 ~]# ext2online /opt/Oracle/

转换好后，让我们查看一下文件系统的当前状态：

[root@li2 ~]# df －lh

Filesystem 1k-blocks Used Available Use% Mounted on

/dev/sda1 7.4G 1.8G 5.3G 25% /

none 135M 0 135M 0% /dev/shm

/dev/mapper/lvmdisk-pldy1 71G 81M 68G 1% /opt/Oracle

### 2．卷组中空间不足

当卷组中没有足够的空间用于扩展逻辑卷的大小时，就需要增加卷组的容量，而增加卷组容量的唯一办法就是向卷组中添加新的物理卷。

首先是增加一块新硬盘（36GB SCSI 硬盘），并对其完成分区、创建物理卷等工作。接下来是利用vgextend命令将新的物理卷（/dev/sdd1）加入到卷组中。

扩展卷组的命令如下：

[root@li2 ~]# vgextend lvmdisk /dev/sdd1

Volume group "lvmdisk" successfully extended

利用vgdisplay命令查看卷组lvmdisk的情况：

[root@li2 ~]# vgdisplay

--- Volume group ---

VG Name lvmdisk

System ID

Format lvm2

Metadata Areas 3

Metadata Sequence No 3

VG Access read/write

VG Status resizable

MAX LV 0

Cur LV 1

Open LV 0

Max PV 0

Cur PV 3

Act PV 3

VG Size 107.97 GB

PE Size 4.00 MB

Total PE 27640

Alloc PE / Size 3840 / 15.00 GB

Free PE / Size 23800 / 92.97 GB

VG UUID l8YPvz-uD7h-oj1A-0qS5-TFcT-mbC7-QbjzCu

完成卷组的扩容后，就可以按照第一种情况的方法完成逻辑卷的扩容，最终实现分区的动态调整。