GNU/Linux
磁盘管理3

逻辑卷管理-LVM

LVM（Logical Volume Manager，逻辑卷管理器）

LVM最早源于IBM的AIX系统

LVM是建立在磁盘和分区之上的一个逻辑层，用来提高磁盘分区管理的灵活性

LVM可以对磁盘分区按照组的方式进行命名、管理和分配

逻辑卷管理-LVM

LVM术语:

1. Physical Volume(PV)

实际分区需要调整System ID成为LVM表示(8e)，然后经过pvcreate命令将他转为LVM最低层的PV,然后才能使用磁盘。

2. Volume Group(VG)

将PV整合成为一起即为VG

逻辑卷管理-LVM

LVM术语:

3. Physical Extent(PE)

LVM预设使用4MB的PE区块，每个LV最多允许有65534个PE，即256GB。

PE属于LVM最小存储区

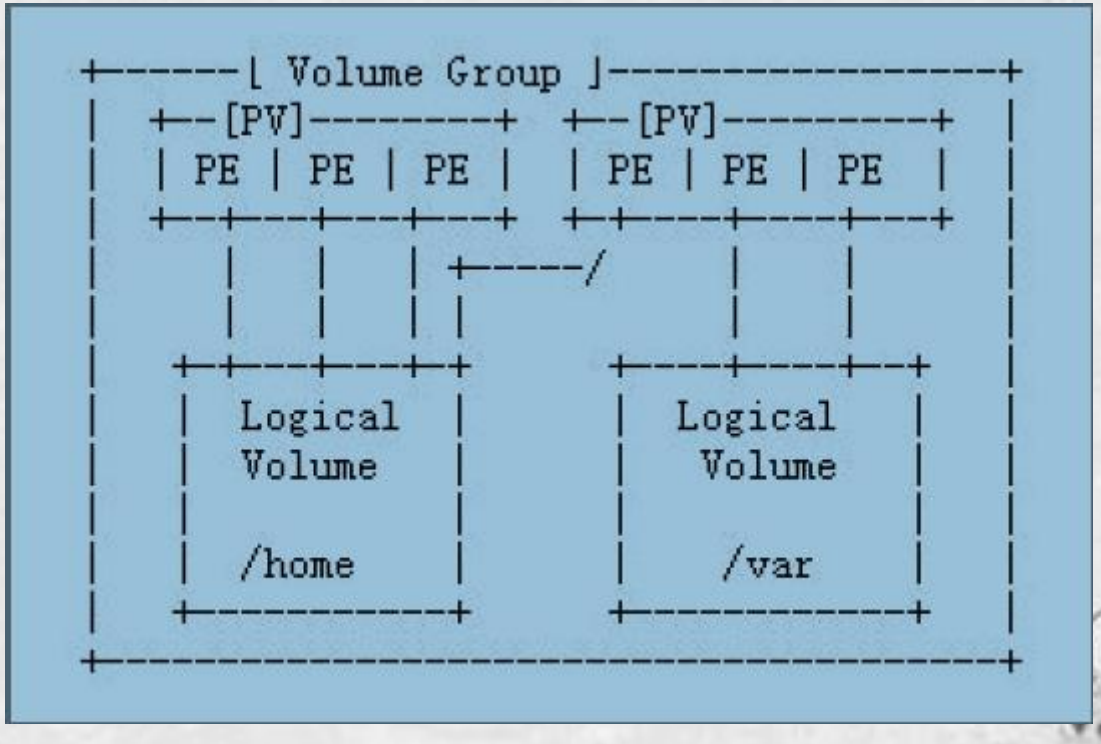
4. Logical Volume(LV)

VG将被分割为若干LV，LV的大小受到PE的限制，且不可以随意调整大小。

逻辑卷管理-LVM

LVM术语:

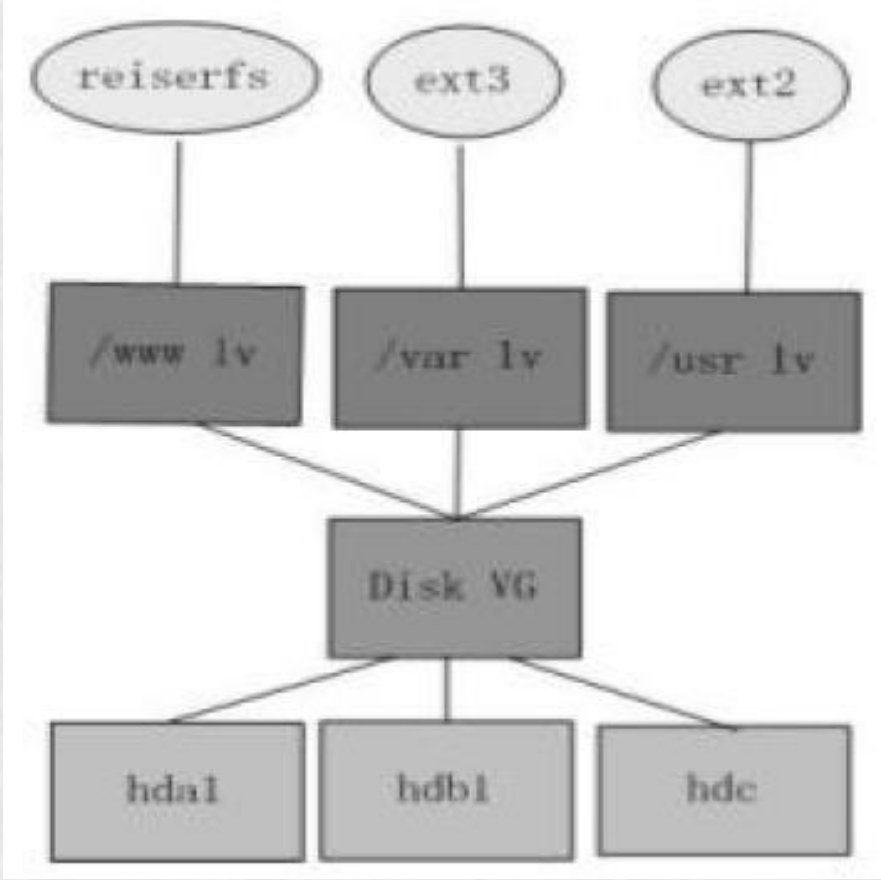
PV/VG/PE/LV/LE关系



逻辑卷管理-LVM

LVM术语:

PV/VG/PE/LV/LE关系



逻辑卷管理-LVM

LVM实现所涉及到的命令:

系统文件分区:mkfs,mount
LV阶段:lvcreate,lvdisplay
VG阶段:vgcreate,vgdisplay
PV阶段:pvcreate,pvscan
分区阶段:fdisk/parted/cfdisk

逻辑卷管理-LVM

LVM实现

1. 准备 3 块磁盘,并将FS的类型改为LVM(8e)

2. 将三块磁盘转换为PV

```
#pvcreate  /dev/sd[b-d]1
```

3.扫描PV

```
#pvscan
```

逻辑卷管理-LVM

LVM实现

4. 查看pv卷详细信息

#pvdisplay

或

#pvs ←查看当前pv信息

逻辑卷管理-LVM

LVM实现

5. 创建VG

#vgcreate -s 指定PE大小(M,G,T) VG名称 PV名称

#vgcreate -s 指定PE大小 -e 指定此VG总PE数 VG名称 PV名称

如

#vgcreate -s 4M snow /dev/sd[b-c]1

#vgcreate -s 4M -e 1000 snow /dev/sd[b-c]1

逻辑卷管理-LVM

LVM实现

6. 扫描VG

#vgscan

7. 查看VG详细信息

#vgdisplay

逻辑卷管理-LVM

LVM实现

8.建立LV

#lvcreate [-L N[mgt] [-n LV名称] VG名称

-L 指定容量，单位是MB,GB,TB

-n LV名称

如

#lvcreate –L 10G –n lisalv snow

或

#lvcreate -l 100%FREE -n lisalv snow

逻辑卷管理-LVM

LVM实现

10. 查看LV的映射

    #ll /dev/snow/lisalv

11.扫描lv卷

    #lvscan

12.查看lv信息

    #lvdisplay

逻辑卷管理-LVM

LVM实现

13.格式化lv

#mkfs.ext4　/dev/snow/lisalv

14.挂载

15.测试

逻辑卷管理-LVM

LVM实现

16.增加LV空间

1)将/dev/sdd1增加到VG为snow的卷组

#vgextend snow /dev/sdd1

#vgdisplay

逻辑卷管理-LVM

LVM实现

16.增加LV空间

2)在10G的基础上增加9G空间至lisalv

#lvextend -L +9G -f -r /dev/snow/lisalv

3) 对lisalv增加至19G空间

#lvextend -L 19G -f -r /dev/snow/lisalv

逻辑卷管理-LVM

LVM实现

17.减少LV空间

1)在19G的基础上减少9G空间至lisalv

#lvreduce -L -9G -f -r /dev/snow/lisalv

2) 对lisalv减少至10G空间

#lvreduce -L 10G -f -r /dev/snow/lisalv

逻辑卷管理-LVM

LVM实现

XFS:

如果是XFS的文件系统格式,需要安装xfsgrops软件包,通过xfs_grows来完成分区的扩展动作

#xfs_grows 挂载点

如

#xfs_grows /mnt/hd

逻辑卷管理-LVM

LVM实现

命令说明:

lvextend:只允许在现有基础上增加空间

lvreduce:只允许再现有基础上减少空间

参数

-f:强制

-r:resizefs,在扩展/缩减空间之后，空间表面上增加，但实际上仍然需要处理block总数，以达到块与空间容量是对应的。因此无论是扩展/缩减空间都需要对块重新resize。

逻辑卷管理-LVM

LVM删除

1. 卸载所使用的lv

2. 先删除lv

```
#lvremove /dev/snow/lisalv
```

3. 移除vg中的pv

```
#vgreduce snow /dev/sdd1
#vgreduce snow /dev/sdc1
```

逻辑卷管理-LVM

LVM删除

4. 从pv中删除指定设备

```
#pvremove  /dev/sdd1
#pvremove  /dev/sdc1
```

5. 删除vg

```
#vgremove snow
```

逻辑卷管理-LVM

LVM快照

通过lvm快照给lvm真实卷拍个照片，当lvm真实卷发送改变时，lvm快照把lvm真实卷改变之前的内容存放在快照上，这样在lvm快照有效的这段时间内，我们看到的lvm快照上的内容始终是lvm真实卷在创建lvm快照时内容，通过备份lvm快照即可达到在线备份lvm真身的目的。

逻辑卷管理-LVM

LVM快照

需要注意的是，当lvm快照比lvm真身小时，若lvm真身发生的改变大于lvm快照，则lvm快照将变得无法读取而失效; 若lvm快照大于等于lvm真身，则不会发生前面的情况。

逻辑卷管理-LVM

LVM快照

前期工作

1. 准备好LV

2. 在LV上进行一些写操作

逻辑卷管理-LVM

创建LVM快照

```
# lvcreate -L 100M -s -n slisalv /dev/snow/lisalv
```

-s:建立快照

-n:快照的LV名称

/dev/snow/lisalv为快照所对应的真实卷

逻辑卷管理-LVM

查看LVM中的LV

```
# lvs
```

挂载LV及快照LV(快照LV不需要建立文件系统)

```
#mkdir -v /mnt/lv
#mount /dev/snow/lisalv /mnt/lv
```

逻辑卷管理-LVM

查看LV真实卷及LV快照内容,确认数据

#cd /mnt/lv

#ll

#cd /mnt/lvsn

#ll

逻辑卷管理-LVM

查看LVM中的LV

```
# lvs
```

挂载快照LV(快照LV不需要建立文件系统)

```
#mkdir -v /mnt/lvsn
#mount /dev/snow/slisalv /mnt/lvsn
```

逻辑卷管理-LVM

快照完成后可直接导出为文档

#dd if=/dev/snow/slisalv of=/backup/lisalv.img

每次需要生成新的快照可以先将旧有的快照删除，而后重新建立快照卷即可

#lvremove /dev/snow/slisalv

lvcreate -L 100M -s -n slisalv /dev/snow/lisalv

逻辑卷管理-LVM

创建条带型LV

1. 创建PV

#pvcreate /dev/sd[a-b]1

2.创建VG

#vgcreate snow_striped /dev/sd[a-b]1

逻辑卷管理-LVM

创建条带型LV

3. 创建条带LV

#lvcreate -l 100%FREE -i 2 -I 64 -n lisa_striped snow_striped

-l : 使用所有的自由的LE(小写L)

-i : 设定制作条带的磁盘数量

-I : 指定多少K字节做一个条带区块

逻辑卷管理-LVM

创建条带型LV

4. 查看条带LV信息

#lvdisplay

及

#lvs -a -o vg_name,name,devices,size

逻辑卷管理-LVM

带有镜像能力的LV

1. 创建一个VG

#vgcreate snowvg_mirror /dev/sd[c-d]1

2. 创建镜像LV

#lvcreate -L 4G -m1 -n lv_mirror vg_mirror

逻辑卷管理-LVM

带有镜像能力的LV

3. 查看镜像LV

#lvdisplay

4. 查看镜像同步信息

#lvs

5.测试

逻辑卷管理-LVM

对现有VG增加带有镜像能力的LV

1. 创建一个普通LV

#pvcreate /dev/sda1

#vgcreate -s 4M snowvg /dev/sda1

#lvcreate -L 4G -n lisalv snowcg

#mkfs.ext4 /dev/snowvg/lisalv

#lvdisplay

逻辑卷管理-LVM

对现有VG增加带有镜像能力的LV

2. 将新的PV加入至现有VG组中

#vgextend snowvg /dev/sdb1

3.

#lvconvert -m1 /dev/snowvg/lisalv /dev/sdb1

逻辑卷管理-LVM

对Mirror-LV的修复

1.对Mirror-LV进行写操作

2. 确认LV当前状态

#lvs -a -o +devices

#lvs

2.对某个PV进行破坏

#dd if=/dev/zero of=/dev/sdb1 count=10

逻辑卷管理-LVM

对Mirror-LV的修复

3.对某个PV进行破坏

#dd if=/dev/zero of=/dev/sdb1 count=10

4. 查看破坏后的状态(sdb1消失)

#lvs -a -o +devices

#lvs

逻辑卷管理-LVM

对Mirror-LV的修复

5.验证Mirror-LV的数据可用性

#umount 挂载点

#mount /dev/snowvg/lisalv /mnt/niliu

6.将损坏的设备移除掉

#vgreduce --removemissing --force snowvg

逻辑卷管理-LVM

对Mirror-LV的修复

7. 解除LV的镜像

#lvconvert -m0 /dev/snowvg/lisalv

8.确认LV的Mirrored Volumes已解除

#lvdisplay

逻辑卷管理-LVM

对Mirror-LV的修复

9.进行数据恢复

#pvcreate /dev/sdc1

#vgextend snowvg /dev/sdc1

#lvconvert -m1 /dev/snowvg/lisalv /dev/sdd1

10.确认Mirrored Volumes存在且数据开始同步

#lvdisplay

#lvs