LVM 故障诊断和排错

[重点]（未解决）# 问题 ：6.1 中提到的 LVM 诊断过程

[答案] ：

1、 检查当前应用是否受损

2、 当前应用是否需要停止

3、 卷组是否正常

4、 逻辑卷是否正常

5、 pv 是否正常？是否有物理磁盘损坏

6、 是否需要恢复 pv ？

7、 是否需要恢复卷组的 meta data ？

==========

（已解决）# 问题 ：lvmdump 命令的用法？

[答案] ：

lvmdump 是一个用于 dump 当前 Lvm 相关信息的工具。

它可以导出如下的内容 ：

-）1、dmsetup info 的输出

-）2、当前运行的进程信息

-）3、/var/log/messages 文件

-）4、LVM 配置文件和缓存文件

-）5、/dev 目录下的文件列表

-）6、如果带 -m 选项，则导出 meta data

-）7、如果带 -a 选项，则导出 pvscan、vgscan 、lvs、pvs、vgs 的 debug 输出以及可用的 vg、pv、lv 列表

-）8、如果带 -c 选项，则导出集群信息

2、lvmdump 默认的输出是一个 tgz 包，名称是 lvmdump-<hostname>-<time>.tgz 。解压可以用 tar -zxvf

3、lvmdump 也可以把输出导出到一个指定的目录下。如果没有该目录，lvmdump 将自动建立。

==========

（已解决）# 问题 ：执行 lvmdump ，并查看它生成的内容

[答案] ：

1、首先执行 lvmdump 命令（注意，只有 root 才能执行，普通用户无法执行）

[root@mail ~]# lvmdump -am

Creating dump directory: /root/lvmdump-mail.bob.com-2007060683747 // 先建立一个临时目录，最后会删掉

Gathering LVM volume info... // 收集 LVM 的信息

vgscan...

pvscan...

lvs...

pvs...

vgs...

Gathering LVM & device-mapper version info... // 收集 device mapper 和 LVM 的版本信息

Gathering dmsetup info... // 收集 dmsetup info 的结果

Gathering process info... // 收集进程信息

Gathering console messages... // 收集 console 的信息

Gathering /etc/lvm info... // 拷贝 lvm.conf

```
Gathering /dev listing...                                    // 收集 /dev 下的文件列表
Gathering LVM metadata from Physical Volumes...              // 收集 meta data 信息
  /dev/hda6
Creating report tarball in /root/lvmdump-mail.bob.com-2007060683747.tgz...   // 最后生成一个 tgz 包
[root@mail ~]#
```

2、现在解压该 tgz 包并查看其内容

```
[root@mail lvmdump-mail.bob.com-2007060683747]# ll
total 132
-rw-r--r--  1 root root 15913 Jun  6 16:37 dev_listing        // 该文件列出了 /dev 目录下的所有文件，就等于执行 ls -l /dev
-rw-r--r--  1 root root   232 Jun  6 16:37 dmsetup_info       // dmsetup info 的输出
-rw-r--r--  1 root root    28 Jun  6 16:37 dmsetup_status     // dmsetup status 的输出
-rw-r--r--  1 root root    35 Jun  6 16:37 dmsetup_table      // dmsetup table 的输出
drwxr-xr-x  4 root root  4096 Jun  6 16:37 lvm                // 该目录下有 archive 和 backup 目录，以及 lvm.conf 和 .cache
-rw-r--r--  1 root root  2392 Jun  6 16:37 lvmdump.log        // lvmdump 的操作日志
-rw-r--r--  1 root root   140 Jun  6 16:37 lvs                // lvs 的输出
-rw-r--r--  1 root root  5295 Jun  6 16:37 messages           // messages 文件
drwxr-xr-x  2 root root  4096 Jun  6 16:37 metadata           // 该目录是用于导出 meta data
-rw-r--r--  1 root root  7467 Jun  6 16:37 ps_info            // 进程信息
-rw-r--r--  1 root root  2697 Jun  6 16:37 pvs                // pvs -av 的输出
-rw-r--r--  1 root root   288 Jun  6 16:37 pvscan             // pvscan 的输出
-rw-r--r--  1 root root   224 Jun  6 16:37 versions           // 含有 LVM 和 device-mapper 的版本信息
-rw-r--r--  1 root root   166 Jun  6 16:37 vgs                // vgs -v 的输出
```

-rw-r--r-- 1 root root 51040 Jun 6 16:37 vgscan // vgscan -vvvv 的输出

[root@mail lvmdump-mail.bob.com-2007060683747]#

==========

（已解决）# 问题 ：lvmdump 在 LVM 系统出现故障时还能用吗？

[答案] ：

砝在卷组出现 pv 丢失的情况下，或者 lv 丢失的情况下并没有用，vgs、lvs 文件并没有内容，也就是没有用 --partial 选项

==========

（已解决）# 问题 ：lvm dumpconfig 命令的用法

[答案] ：

lvm dumpconfig 是导出 LVM 的配置，和具体的 VG ，lv、pv 是无关的。

[root@mail ~]# lvm dumpconfig

devices {

dir="/dev"

scan="/dev"

filter="a/.*/"

cache="/etc/lvm/.cache"

write_cache_state=1

sysfs_scan=1

md_component_detection=1

ignore_suspended_devices=0

}

activation {

missing_stripe_filler="/dev/ioerror"

```
        reserved_stack=256
        reserved_memory=8192
        process_priority=-18
        mirror_region_size=512
        mirror_log_fault_policy="allocate"
        mirror_device_fault_policy="remove"
}
global {
        umask=63
        test=0
        activation=1
        proc="/proc"
        locking_type=1
        fallback_to_clustered_locking=1
        fallback_to_local_locking=1
        locking_dir="/var/lock/lvm"
}
shell {
        history_size=100
}
backup {
        backup=1
        backup_dir="/etc/lvm/backup"
```

```
        archive=1
        archive_dir="/etc/lvm/archive"
        retain_min=10
        retain_days=30
}
log {
        verbose=0
        syslog=1
        overwrite=0
        level=0
        indent=1
        command_names=0
        prefix="  "
}
[root@mail ~]#
================================================================================
```

[重点] （已解决）# 问题 ：总结恢复 vg 的过程

[答案] ：

我们分成四种情况来看 ：

1）文件系统正常，但 lv 不正常，且 pv 无物理故障。

2）文件系统正常（superblock 没有坏），但 pv 故障（lvm label 或者 metadata 丢失）

3）文件系统不一致（superblock 损坏），但 pv 正常

4）文件系统和 pv 都不正常

注 ：这里讨论的故障都是 “软”故障，而不是真正的物理损坏。

下面分成三种情况讨论

==========

[重点] （已解决）# 问题 ：文件系统正常，但 lv 不正常，且 pv 无物理故障

[答案] ：

-）1、首先备份该 pv 上的文件系统（虽然 pv 不正常，但文件系统还是可以正常挂载的，最好是以 ro 方式挂载）

-）2、最好卸载 vg 和相关的文件系统。尤其是要 deactive 卷组，否则后面的 pvcreate 可能无法执行

-）3、执行 lvs 查看是那个 lv 不正常（一般也是 pv 的 metadata 出现问题才会造成 lv 不正常的）

-）4、用 pvcreate 恢复该 pv （用于恢复的 pv 就是本身）。命令是

[root@mail ~]# pvs -P

Partial mode. Incomplete volume groups will be activated read-only.

Couldn't find device with uuid 'dOt94R-RS3A-NWb6-oCjR-DI8u-BkMI-rOjKNC'.

（省略）

PV VG Fmt Attr PSize PFree

（省略）

unknown device vg_1 lvm2 a- 964.00M 884.00M // 原来是 /dev/hdb5 的，现在变成 unknown 设备了。

[root@mail ~]#

[root@mail ~]# pvcreate -ff --uuid dOt94R-RS3A-NWb6-oCjR-DI8u-BkMI-rOjKNC --restorefile /etc/lvm/backup/vg_1 /dev/hdb5 // 注意到目的 pv 就是 /dev/hdb5 本身

Couldn't find device with uuid 'dOt94R-RS3A-NWb6-oCjR-DI8u-BkMI-rOjKNC'.

Physical volume "/dev/hdb5" successfully created // 提示成功恢复

[root@mail ~]#

[root@mail ~]# pvs

PV	VG	Fmt	Attr	PSize	PFree
/dev/hda6	vg_1	lvm2	a-	964.00M	940.00M
/dev/hda7	vg_2	lvm2	a-	1.87G	808.00M
/dev/hda8		lvm2	--	1.01G	1.01G
/dev/hdb5	vg_1	lvm2	a-	964.00M	884.00M
/dev/hdb6	vg_1	lvm2	a-	3.73G	2.03G

[root@mail ~]#

-）5、现在可以恢复卷组了

[root@mail ~]# vgcfgrestore -f /etc/lvm/backup/vg_1 vg_1

Restored volume group vg_1

[root@mail ~]#

-）6、执行 lvs 查看 lv 的情况

[root@mail ~]# vgchange -ay vg_1

4 logical volume(s) in volume group "vg_1" now active

[root@mail ~]# lvs vg_1

```
  LV     VG    Attr     LSize    Origin Snap%  Move Log         Copy%
  lvol0 vg_1 -wi-a- 800.00M
  lvol1 vg_1 -wi-a- 800.00M
  lvol2 vg_1 owi-a-  40.00M
  lvol3 vg_1 mwi-a-  80.00M                    lvol3_mlog  80.00
  lvol4 vg_1 swi-a-  40.00M lvol2   57.99
[root@mail ~]#
```

可以看到一切正常了，不过建议还是用 fsck 查一遍

==========

[重点] （已解决）# 问题 ：如果是文件系统正常，但 pv 不正常（需要更换的情况）

[答案] ：

和上面一样，但差别有 3 点 ：

-1）、 pvcreate 时目标 pv 不同。

-）2、 pvcrate 时卷组不用 deactive ，因为此时 pv 不属于该 vg

-）3、 pvcreate 用于恢复的 pv 必须是空闲的（不属于任何 vg），大小必须至少等于源 pv，不能小于。否则会报错

-）4、 在恢复卷组后必须用

==========

[重点] （已解决）# 问题 ：文件系统不正常，但 lv ， pv 正常

[答案] ：

这个不关 lvm 的事，用 fsck 修复就是了。但出于 fsck 的风险性，建议在 fsck 之前用 dd 作一下整个 lv 的备份。

-）1、下面是 lv 的情况

```
 [root@mail ~]# lvs
   LV     VG    Attr     LSize    Origin Snap%  Move Log         Copy%
```

```
  lvol0 vg_1 -wi-a- 800.00M
  lvol1 vg_1 -wi-a- 800.00M                                        // 该 lv 必须是 Unmount 的状态，否则无法 fsck
  lvol2 vg_1 owi-a-  40.00M
  lvol3 vg_1 mwi-a-  80.00M                    lvol3_mlog 100.00
  lvol4 vg_1 swi-a-  40.00M lvol2    57.99
  lvol0 vg_2 owi-a-  40.00M
  lvol1 vg_2 swi-a-  40.00M lvol0     0.04
  lvol2 vg_2 -wi-a-   1.00G
[root@mail ~]#
```

-）2、用 dd 命令做 lv 级别的备份

```
[root@mail ~]# dd if=/dev/vg_1/lvol1 of=lvol1.iso
1638400+0 records in
1638400+0 records out
[root@mail ~]#
[root@mail ~]# ll -h lvol1.iso
-rw-r--r--  1 root root 800M Jun 17 21:49 lvol1.iso            // ISO 文件的大小也是 800MB
[root@mail ~]#
```

-）3、用 e2fsck 修复。在真正修复之前，先用 -n 参数观察一下。

```
[root@mail ~]# e2fsck -n /dev/vg_1/lvol1
e2fsck 1.35 (28-Feb-2004)
Couldn't find ext2 superblock, trying backup blocks...
Resize inode not valid.  Recreate? no
```

test was not cleanly unmounted, check forced.

Pass 1: Checking inodes, blocks, and sizes

Inode 7, i_blocks is 1576, should be 1024. Fix? no

Pass 2: Checking directory structure

Pass 3: Checking directory connectivity

Pass 4: Checking reference counts

Pass 5: Checking group summary information

test: ********** WARNING: Filesystem still has errors **********

test: 11/102592 files (9.1% non-contiguous), 7532/204800 blocks

[root@mail ~]#

-）4、真正用 e2fsck 修复

[root@mail ~]# e2fsck -y /dev/vg_1/lvol1

e2fsck 1.35 (28-Feb-2004)

Couldn't find ext2 superblock, trying backup blocks...

Resize inode not valid. Recreate? yes

test was not cleanly unmounted, check forced.

Pass 1: Checking inodes, blocks, and sizes

Pass 2: Checking directory structure

Pass 3: Checking directory connectivity

Pass 4: Checking reference counts

Pass 5: Checking group summary information

Free blocks count wrong for group #0 (28148, counted=28149).

Fix? yes

Free blocks count wrong (197267, counted=197268).

Fix? yes

test: ***** FILE SYSTEM WAS MODIFIED *****

test: 11/102592 files (0.0% non-contiguous), 7532/204800 blocks

[root@mail ~]#

-）5、用 fsck 再次检查，已经正常了

[root@mail ~]# e2fsck /dev/vg_1/lvol1

e2fsck 1.35 (28-Feb-2004)

test: clean, 11/102592 files, 7532/204800 blocks

[root@mail ~]#

==========