Distributed　Logical Volume with Striped

环境：本文仅讨论实现，并不涉及性能和安全

IP-SAN(iscsi targets) four server

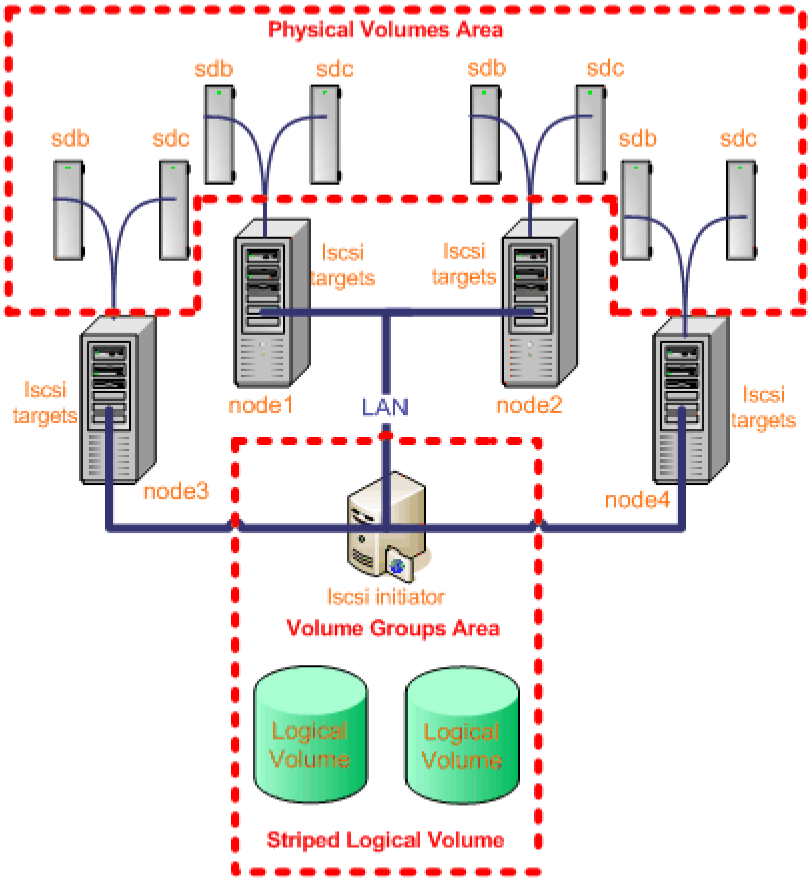
One clients

平台：Vmware 6.0 ACE

CentOS5 update 2 x86

我的 4 个节点分别为 gfs1 gfs2 gfs3 gfs4

网络拓扑图：



Distributed Logical Volume with Striped

一、前言

首先我们做一个试验：

```
[root@gfs2 ~]# mkfs.ext3 /dev/sdb
mke2fs 1.39 (29-May-2006)
/dev/sdb is entire device, not just one partition!
```

```
Proceed anyway? (y,n) y
Filesystem label=
OS type: Linux
Block size=4096 (log=2)
Fragment size=4096 (log=2)
1310720 inodes, 2621440 blocks
131072 blocks (5.00%) reserved for the super user
First data block=0
Maximum filesystem blocks=2684354560
80 block groups
32768 blocks per group, 32768 fragments per group
16384 inodes per group
Superblock backups stored on blocks:
        32768, 98304, 163840, 229376, 294912, 819200, 884736, 1605632
Writing inode tables: done
Creating journal (32768 blocks): done
Writing superblocks and filesystem accounting information: done
This filesystem will be automatically checked every 20 mounts or
180 days, whichever comes first.  Use tune2fs -c or -i to override.
[root@gfs2 ~]# mkdir /test
[root@gfs2 ~]# mount /dev/sdb /test
[root@gfs2 ~]# df -h
Filesystem            Size  Used Avail Use% Mounted on
/dev/mapper/VolGroup00-LogVol00
                      7.0G  2.1G  4.5G  32% /
/dev/sda1              99M   12M   83M  13% /boot
tmpfs                 189M     0  189M   0% /dev/shm
/dev/hdc              3.8G  3.8G     0 100% /media/CentOS_5.2_Final
/dev/sdb              9.9G  151M  9.2G   2% /test
```

做这个试验的目的是要大家知道，一块磁碟不一定要分区。一样能使用。但是在测试中 linux 的软 RAID 一定要分区。现在回到主机 gfs1 上去做 LVM 实验。

二、建立一个单级条带的逻辑卷

1、建立 Physical Volumes

```
[root@gfs1 ~]# pvcreate /dev/sdb
  Physical volume "/dev/sdb" successfully created
[root@gfs1 ~]# pvcreate /dev/sdc
  Physical volume "/dev/sdc" successfully created
```

2、建立 Volume Group

```
[root@gfs1 ~]# vgcreate VG0 /dev/sdb /dev/sdc
  Volume group "VG0" successfully created
```

3、建立一个单级条带的逻辑卷

```
[root@gfs1 ~]# lvcreate -L 2G -n lv0 VG0
  Logical volume "lv0" created
```

4、用 gfs 格式化逻辑卷

```
[root@gfs1 ~]# [root@gfs1 ~]# gfs_mkfs -p lock_nolock -j 1 /dev/VG0/lv0
This will destroy any data on /dev/VG0/lv0.
Are you sure you want to proceed? [y/n] y
Blocksize:                  4096
-bash: This: command not found
Filesystem Size:            491460
Journals:                   1

Locking Protocol:           lock_nolock
Lock Table:
Syncing...
All Done
```

1.5 挂载格式化完成的文件系统

```
[root@gfs1 ~]# mkdir /gfs_nolock
```

```
[root@gfs1 ~]# mount -t gfs /dev/VG0/lv0 /gfs_nolock/
[root@gfs1 ~]# df -h
Filesystem            Size  Used Avail Use% Mounted on
/dev/mapper/VolGroup00-LogVol00
                      7.0G  2.1G  4.5G  32% /
/dev/sda1              99M   12M   83M  13% /boot
tmpfs                 189M     0  189M   0% /dev/shm
/dev/mapper/VG0-lv0   1.9G   20K  1.9G   1% /gfs_nolock
[root@gfs1 ~]# mount -l -t gfs
/dev/mapper/VG0-lv0      on      /gfs_nolock      type      gfs
(rw,localflocks,localcaching,oopses_ok)
```

三、建立一个多级条带的逻辑卷

1、建立 Physical Volumes

```
[root@gfs3 ~]# pvcreate /dev/sd[b,c]
  Physical volume "/dev/sdb" successfully created
  Physical volume "/dev/sdc" successfully created
```

2、建立 Volume Group

```
[root@gfs3 ~]# vgcreate vg0 /dev/sdb /dev/sdc
  Volume group "vg0" successfully created
```

3、建立一个多级条带的逻辑卷

```
[root@gfs3 ~]# lvcreate -i2 -I4 -L3G -nlv0 vg0
  Logical volume "lv0" created
```

4、用 gfs 格式化逻辑卷

```
[root@gfs3 ~]# gfs_mkfs -plock_nolock -j 1 /dev/vg0/lv0
This will destroy any data on /dev/vg0/lv0.
Are you sure you want to proceed? [y/n] y
Device:                    /dev/vg0/lv0
Blocksize:                 4096
```

```
Filesystem Size:            753580
Journals:                     1
Resource Groups:              12
Locking Protocol:           lock_nolock
Lock Table:
Syncing...
All Done
```

4、挂载格式化完成的文件系统

```
[root@gfs3 ~]# mkdir /testlv
[root@gfs3 ~]# mount -t gfs /dev/vg0/lv0 /testlv/
[root@gfs3 ~]# df -h
Filesystem            Size  Used Avail Use% Mounted on
/dev/mapper/VolGroup00-LogVol00
                      7.0G  2.1G  4.5G  32% /
/dev/sda1              99M   12M   83M  13% /boot
tmpfs                 189M     0  189M   0% /dev/shm
/dev/hdc              3.8G  3.8G     0 100% /media/CentOS_5.2_Final
/dev/mapper/vg0-lv0   2.9G   20K  2.9G   1% /testlv
[root@gfs3 ~]# mount -l -t gfs
/dev/mapper/vg0-lv0 on /testlv type gfs (rw,localflocks,localcaching,oopses_ok)
```

其他的一些操作我就不再多讲了。网上很多大家可以搜索。现在讲下和 IP-SAN 的具体的分布式的应用。大家都知道 LUSTRE 是个分布式的集群文件系统。其实 GFS 本身不是完全分布式的。（这里过气的 GBIND 就不再讨论）他仅仅是一个有 LOCK 机制和多 journal 的文件系统。靠分布式的是它下层的 LVM。大家可能看过红帽的 RHCS 的 OVERVIEW 里面有个讲 LVM2 的 CLUSTER 的图片。相信大家都熟悉 clvmd 这个程式。这个程式运行在 GFS 的 node 上。这个程式的作用仅仅是能让 GFS node 识别 share storage 上的逻辑卷。其实和分布式没有任何一点关系。下面我就尝试一下用 ISCSI+LVM 来逻辑分布存储，如果用单

级条带分布的话，其实没有任何意义。LV 的 I/O 也上不去。

四、Distributed　Logical Volume with Striped

1、在4个节点上先把本地磁盘target出来

```
[root@gfs1 ~]# yum install scsi-target-utils
[root@gfs1 ~]# chkconfig tgtd on
[root@gfs1 ~]# service tgtd restart
Stopping SCSI target daemon:
Starting SCSI target daemon:                                    [  OK  ]
```

定义 target 的 qualified 的名字

```
[root@gfs1 ~]# tgtadm --lld iscsi --op new --mode target --tid 1 -T iqn.2008-12.sys.sdb
[root@gfs1 ~]# tgtadm --lld iscsi --op new --mode target --tid 2 -T iqn.2008-12.sys.sdc
```

为创建目标增加分区

```
[root@gfs1 ~]# tgtadm --lld iscsi --op new --mode logicalunit --tid 1 --lun 1 -b /dev/sdb
[root@gfs1 ~]# tgtadm --lld iscsi --op new --mode logicalunit --tid 2 --lun 1 -b /dev/sdc
```

定义客户端的访问

```
[root@gfs1 ~]# tgtadm --lld iscsi --op bind --mode target --tid 1 -I ALL
[root@gfs1 ~]# tgtadm --lld iscsi --op bind --mode target --tid 2 -I ALL
```

验证

```
[root@gfs1 ~]# tgtadm --lld iscsi --op show --mode target |grep Target
Target 1: iqn.2008-12.sys.sdb
Target 2: iqn.2008-12.sys.sdc
```

到这里我的其他 4 台机器都一样。所以我搞个脚本去运行就可以了。我这里是为了图简便。希望如果你要有什么价值的应用的话。自己理顺一下每个节点的 target qualified 的名字。

2、调整 client 端，发现：

```
[root@client ~]# iscsiadm -m discovery -t sendtargets -p gfs1
172.18.174.1:3260,1 iqn.2008-12.sys.sdb
172.18.174.1:3260,1 iqn.2008-12.sys.sdc
[root@client ~]# iscsiadm -m discovery -t sendtargets -p gfs2
172.18.174.2:3260,1 iqn.2008-12.sys.sdb
172.18.174.2:3260,1 iqn.2008-12.sys.sdc
[root@client ~]# iscsiadm -m discovery -t sendtargets -p gfs3
172.18.174.3:3260,1 iqn.2008-12.sys.sdb
172.18.174.3:3260,1 iqn.2008-12.sys.sdc
[root@client ~]# iscsiadm -m discovery -t sendtargets -p gfs4
172.18.174.4:3260,1 iqn.2008-12.sys.sdb
172.18.174.4:3260,1 iqn.2008-12.sys.sdc
```

```
[root@client ~]# service iscsi restart
```

完成后你就可以去数盘了。哈哈我的是：sd[b,c,d,e,f,g,h,i]八个。

```
Disk /dev/sdb: 10.7 GB, 10737418240 bytes
Disk /dev/sdc: 17.1 GB, 17179869184 bytes
Disk /dev/sdd: 10.7 GB, 10737418240 bytes
Disk /dev/sdf: 10.7 GB, 10737418240 bytes
Disk /dev/sdg: 17.1 GB, 17179869184 bytes
Disk /dev/sdh: 17.1 GB, 17179869184 bytes
Disk /dev/sde: 17.1 GB, 17179869184 bytes
Disk /dev/sdi: 10.7 GB, 10737418240 bytes
```

这里可以看下 initiator 这端的标示是不规则的。都是靠 UDEV 来扫描生成盘符。所以你可以调整 UDEV 让其固定盘符。我这里测试就不固定了。最后我分两个 VG 。容量一样的分到一个 VG（为什么，自己做下实验不一样的分下就知道了）

```
[root@client ~]# pvcreate /dev/sd{b,c,d,e,f,g,g,i} -ff
  /dev/cdrom: open failed: Read-only file system
  Attempt to close device '/dev/cdrom' which is not open.
  Physical volume "/dev/sdb" successfully created
  Physical volume "/dev/sdc" successfully created
  Physical volume "/dev/sdd" successfully created
  Physical volume "/dev/sde" successfully created
  Physical volume "/dev/sdf" successfully created
  Physical volume "/dev/sdg" successfully created
  Physical volume "/dev/sdg" successfully created
Really INITIALIZE physical volume "/dev/sdi" of volume group "vg0" [y/n]? y
  WARNING: Forcing physical volume creation on /dev/sdi of volume group "vg0"
  Physical volume "/dev/sdi" successfully created
[root@client ~]# vgcreate iscsi_vg_10g /dev/sd{b,d,f,i}
  Volume group "iscsi_vg_10g" successfully created
[root@client ~]# vgcreate iscsi_vg_17g /dev/sd{c,g,h,e}
  Volume group "iscsi_vg_17g" successfully created
[root@client ~]# lvcreate -i4 -I4 -l10236 -n10g_lv iscsi_vg_10g
  /dev/cdrom: open failed: Read-only file system
  Logical volume "10g_lv" created

[root@client ~]# lvcreate -i4 -I4 -l16380 -n17g_lv iscsi_vg_17g

/dev/cdrom: open failed: Read-only file system
  Logical volume "17g_lv" created
```

格式化：

```
[root@client ~]# mkfs.ext3 /dev/iscsi_vg_10g/10g_lv
mke2fs 1.40.11 (17-June-2008)
Filesystem label=
OS type: Linux
Block size=4096 (log=2)
Fragment size=4096 (log=2)
5242880 inodes, 10481664 blocks
```

```
524083 blocks (5.00%) reserved for the super user
First data block=0
Maximum filesystem blocks=0
320 block groups
32768 blocks per group, 32768 fragments per group
16384 inodes per group
Superblock backups stored on blocks:
        32768, 98304, 163840, 229376, 294912, 819200, 884736, 1605632,
2654208,
        4096000, 7962624

Writing inode tables: done
Creating journal (32768 blocks): done
Writing superblocks and filesystem accounting information: done

This filesystem will be automatically checked every 28 mounts or
180 days, whichever comes first.  Use tune2fs -c or -i to override.
```

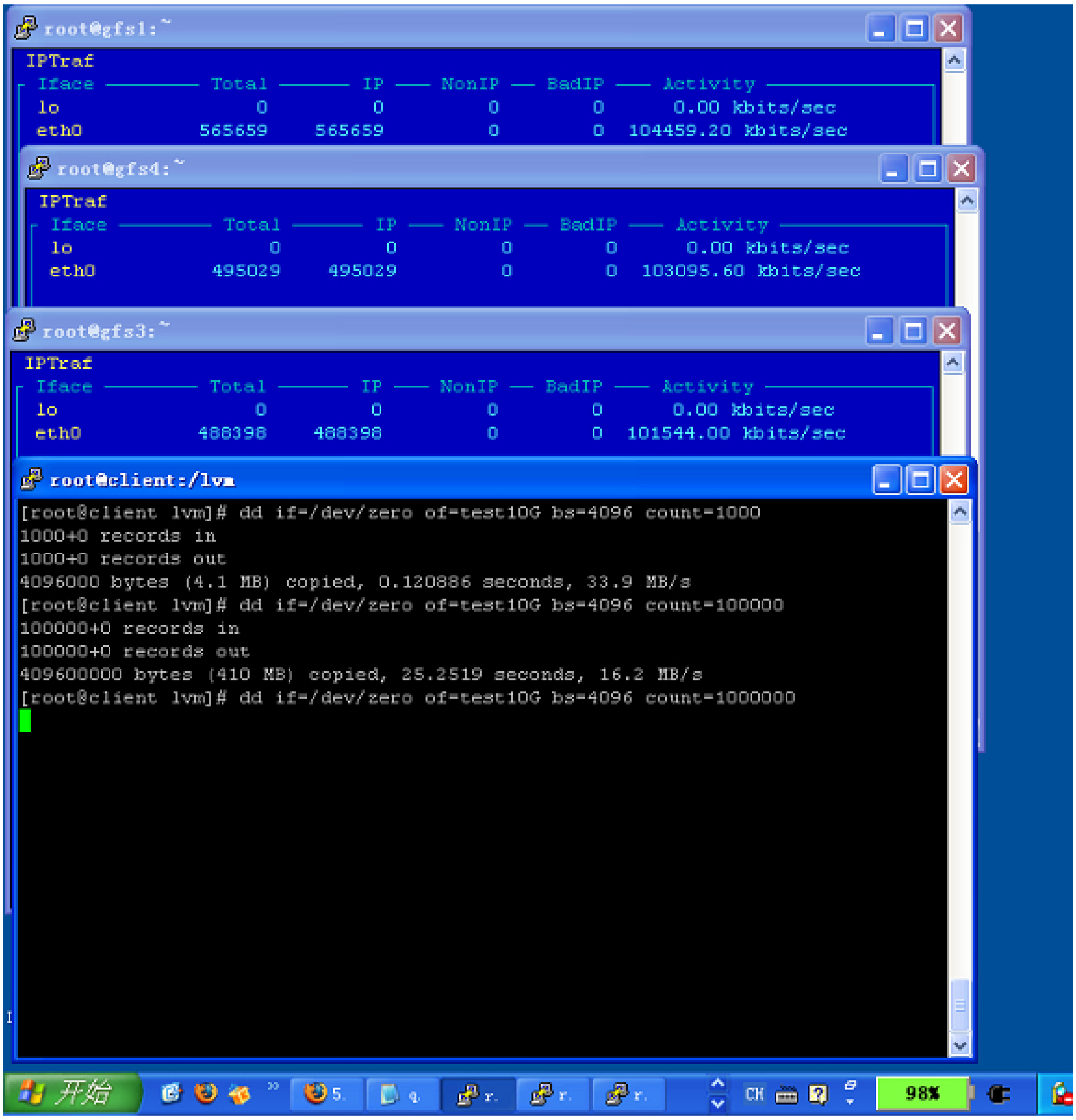
五、测试

我这里格式化一个做测试就够了。

```
[root@client /]# mkdir lvm
[root@client /]# mount /dev/iscsi_vg_10g/10g_lv /lvm
[root@client /]# df -h
Filesystem            Size  Used Avail Use% Mounted on
/dev/mapper/VolGroup00-LogVol00
                      7.2G  2.2G  4.7G  32% /
/dev/sda1              99M   18M   77M  19% /boot
tmpfs                  62M     0   62M   0% /dev/shm
/dev/mapper/iscsi_vg_10g-10g_lv
```

```
            40G   177M    38G     1% /lvm
```

4 个 ISCSI TARGETS 的节点都开启终端。用 iptarf 监控流量。由于屏幕的原因我只监视了 3 个节点。（最后再申明我只是一个实验，没有涉及性能和安全。）



```
IPTraf
┌ Iface ─────────── Total ──────── IP ─── NonIP ── BadIP ─── Activity ──────────────┐
  lo                    0            0        0         0        0.00 kbits/sec
  eth0             565659       565659        0         0   104459.20 kbits/sec
```

```
IPTraf
┌ Iface ─────────── Total ──────── IP ─── NonIP ── BadIP ─── Activity ──────────────┐
  lo                    0            0        0         0        0.00 kbits/sec
  eth0             495029       495029        0         0   103095.60 kbits/sec
```

```
IPTraf
┌ Iface ─────────── Total ──────── IP ─── NonIP ── BadIP ─── Activity ──────────────┐
  lo                    0            0        0         0        0.00 kbits/sec
  eth0             488398       488398        0         0   101544.00 kbits/sec
```

```
[root@client lvm]# dd if=/dev/zero of=test10G bs=4096 count=1000
1000+0 records in
1000+0 records out
4096000 bytes (4.1 MB) copied, 0.120886 seconds, 33.9 MB/s
[root@client lvm]# dd if=/dev/zero of=test10G bs=4096 count=100000
100000+0 records in
100000+0 records out
409600000 bytes (410 MB) copied, 25.2519 seconds, 16.2 MB/s
[root@client lvm]# dd if=/dev/zero of=test10G bs=4096 count=1000000
```

(qq174375@gmail.com)