Lustre with the IP-SAN

环境：本文仅讨论实现，并不涉及性能和安全

IP-SAN(iscsi targets) one server

Two Metadata Servers

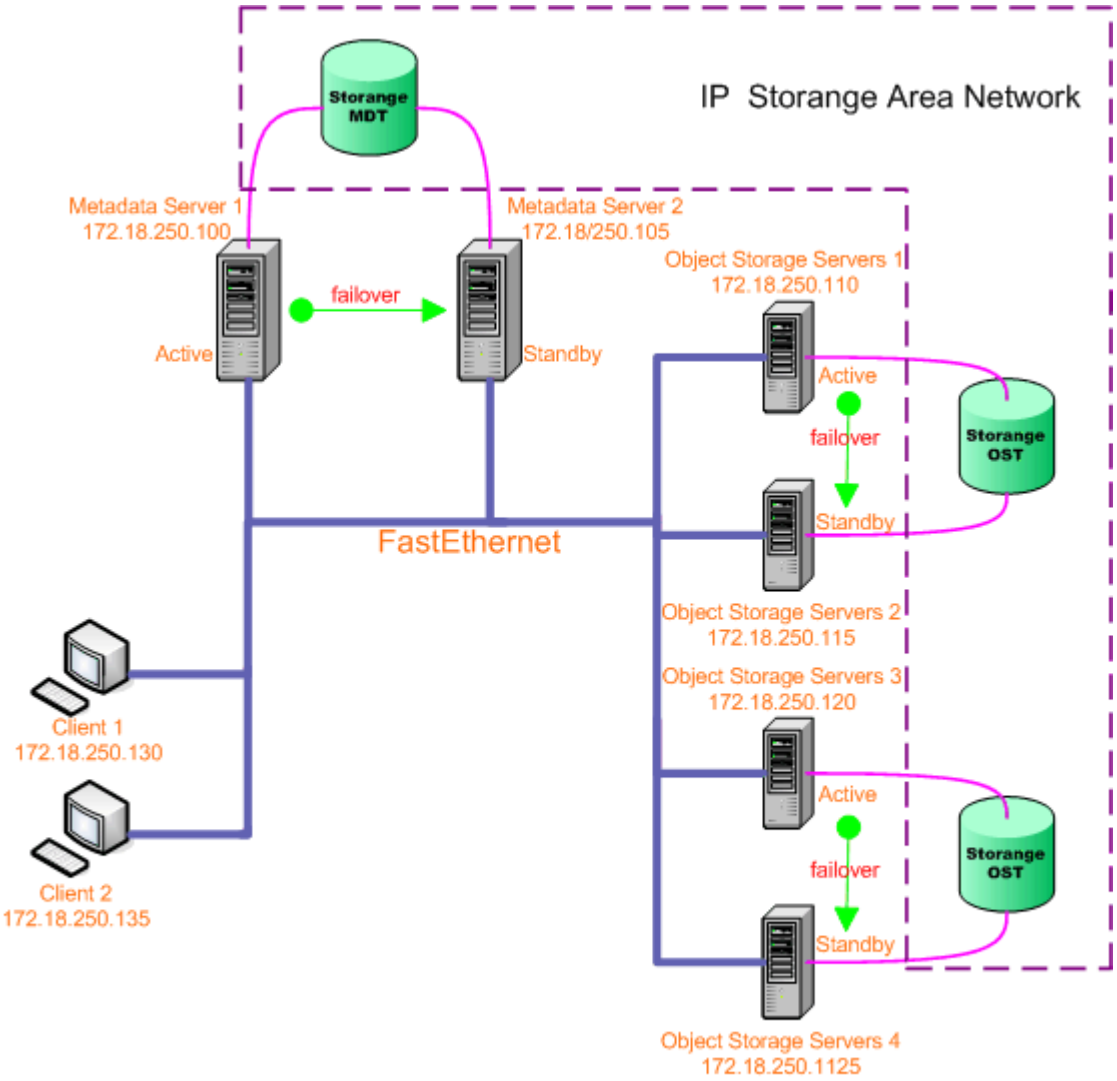
Four Object Storage Servers

Two clients

平台：Vmware 6.0 ACE

CentOS5 update 2 x86

网络拓扑：



Lustre with the IP-SAN

由于需要用到 iscsi 但是 SUN 的原始的 rpm kernel 里没有带有支持 iscsi Initiator，

可能现在 Lustre 根本就看不上 ISCSI 这个玩意吧，所以开始我们需要做一些准备工作自己编译一次内核，让其支持 ISCSI。为什么不选择用*.tar.gz 的包来装的原因就是我嫌弃难得打补丁，自己习惯。愿意用哪个就哪个吧。

一、平台部署

1、安装 kernel 源代码

```
rpm -ivh kernel-lustre-source-2.6.18-92.1.10.el5_lustre.1.6.6.i686.rpm
```

2、安装 expect

```
yum install expect
```

3、安装 lustre 的源代码

```
rpm -ivh lustre-source-1.6.6-2.6.18_92.1.10.el5_lustre.1.6.6smp.i686.rpm
```

4、准备编译 kernel

```
cd /usr/src/linux-2.6.18-92.1.10.el5_lustre.1.6.6
```

这里需要注意的是下面我使用的 config-2.6.18-92.1.10.el5_lustre.1.6.6smp 这个文件就是 SUN 带的 kernel 里的文件，想办法把他弄出来吧。什么方式都可以可以用 cpio 转，也可以用 7-zip 直接解压缩。总之你觉得什么方便就用什么，最后把这个文件拷贝到我们需要编译的 kernel source 下，使用我们拷贝过来的配置文件做为我们编译的基础：

```
cp ~/config-2.6.18-92.1.10.el5_lustre.1.6.6smp .config
make oldconfig || make menuconfig
```

5、内核加入 ISCSI 的支持

```
make menuconfig
Device Driver--SCSI device support--SCSI Transport--iSCSI Transport Attributes
Device Driver--SCSI device support--SCSI low-level drivers-- iSCSI Initiator over TCP/IP
```

6、编译及安装内核

```
make dep
make clean
make -j 8 bzImage
make -j 8 modules
```

make -j 8 modules_install

depmod -a

make install

安装完毕，为了其他的机器好直接用我们把编译的内核生成 RPM 的文件，但是很郁闷的是我的生成后没有 initrd-*.img 这个文件。还要手动写 grub.conf。不过这些都不是问题有模块就可以了。mkinitrd 一个吧，我这里使用的是虚拟机克隆，所以没用到。但是我测试过。编译好后的 rpm 可以正常使用，只是需要自己手工去捣鼓一下。就当是做一次 RHCE 的上午 TS 的一道题吧

make -j 8 rpm

7、编译 lustre 并生成 rpm 包

cd lustre-1.6.6/

./configure --with-linux=/usr/src/linux-2.6.18-92.1.10.el5_lustre.1.6.6

make rpms

8、编辑 grub.conf 确定启动默认是 Lustre 的内核

9、重启

reboot

10、安装 e2fsprogs（我是源代码装的）

tar zxvf e2fsprogs-1.40.11.tar.gz

./configure

make

make install

11、安装 Lustre

安装 lustre 的包(我安装的全是我自己编译出来的包)

rpm -ivh lustre-modules-1.6.6-2.6.18_92.1.10.el5_lustre.1.6.6custom_200812171613.i386.rpm

rpm -ivh lustre-ldiskfs-3.0.6-2.6.18_92.1.10.el5_lustre.1.6.6custom_200812171615.i386.rpm

rpm -ivh lustre-1.6.6-2.6.18_92.1.10.el5_lustre.1.6.6custom_200812171613.i386.rpm

二、IP 及 IP-SAN 规划

1、规划 ISCSI

由于 ISCSI inititor 启动的时候顺序不一。所以在 targets 上一定要规划好

```
root@iscsi:~# grep -v "#" /etc/ietd.conf |grep -v ^$
Target iqn.2008-12.cn.test:lustre.sdb.mgt12.xyz
        Lun 0 Path=/dev/sdb1,Type=fileio
Target iqn.2008-12.cn.test:lustre.sdc.ost12.xyz
        Lun 0 Path=/dev/sdc1,Type=fileio
Target iqn.2008-12.cn.test:lustre.sdd.ost34.xyz
        Lun 0 Path=/dev/sdd1,Type=fileio
root@iscsi:~# grep -v "#" /etc/initiators.deny |grep -v ^$
ALL ALL
root@iscsi:~# grep -v "#" /etc/initiators.allow |grep -v ^$
iqn.2008-12.cn.test:lustre.sdd.ost34.xyz 172.18.250.100, 172.18.250.105
iqn.2008-12.cn.test:lustre.sdc.ost12.xyz 172.18.250.110, 172.18.250.115
iqn.2008-12.cn.test:lustre.sdb.mgt12.xyz 172.18.250.120, 172.18.250.125
```

2、IP 规划

我 IP 规划。自己在 hosts 文件里定义好

```
root@iscsi:~# grep -v "#" /etc/hosts |grep -v ^$
127.0.0.1       localhost
172.18.250.100  mds1
172.18.250.105  mds2
172.18.250.110  oss1
172.18.250.115  oss2
172.18.250.120  oss3
172.18.250.125  oss4
172.18.250.130  client1
172.18.250.135  client2
172.18.250.250  iscsi
```

3、安装 iscsi initiator

```
yum install "*iscsi*"
```

chkconfig iscsid on

chkconfig iscsi on

4、配置 Lustre 的网络及模块

在/etc/modprobe.conf 添加一行

options lnet networks=tcp

这个时候前期工作基本上就做完了。现在需要做的是装剩余的机器。当然我的 ISCSI 是单独做的。我做了一台 mds1 来克隆。克隆完剩余的 mds2 oss1 oss2 oss3 oss4 client1 client2 后需要修改的是：

第一：主机名 /etc/sysconfig/network

第二：IP 地址 /etc/sysconfig/network-scripts/下面的对应文件

然后确定你的环境正确后在 mds1 mds2 oss1 oss2 oss3 oss4 都挂载 ISCSI

iscsiadm -m discovery -t sendtargets -p iscsi

service iscsi restart

fdisk -l

看下挂载上没。我的环境里都出现了一个/dev/sdb

三、配置 Lustre 及其 Failover

1、MDS 的配置

[root@mds1 ~]# mkfs.lustre --fsname=testfs --mdt --mgs --failnode=mds2 /dev/sdb

[root@mds1 ~]# mkdir -p /mnt/mdt

[root@mds1 ~]# mount -t lustre /dev/sdb /mnt/mdt

[root@mds2 ~]# mkdir -p /mnt/mdt

[root@mds2 ~]# mount -t lustre /dev/sdb /mnt/mdt

2、OSS 的配置

[root@oss1 ~]mkfs.lustre --fsname=testfs --ost --failnode=oss2 --mgsnode=mds1 --mgsnode=mds2 /dev/sdb

[root@oss1 ~]# mkdir -p /mnt/ost

[root@oss1 ~]# mount -t lustre /dev/sdb /mnt/ost

[root@oss2 ~]# mkdir -p /mnt/ost

[root@oss2 ~]# mount -t lustre /dev/sdb /mnt/ost

[root@oss3 ~]mkfs.lustre --fsname=testfs --ost --failnode=oss4 --mgsnode=mds1 --mgsnode=mds2 /dev/sdb

[root@oss3 ~]# mkdir -p /mnt/ost

[root@oss3 ~]# mount -t lustre /dev/sdb /mnt/ost

[root@oss4 ~]# mkdir -p /mnt/ost

[root@oss4 ~]# mount -t lustre /dev/sdb /mnt/ost

3、Clients 的配置

[root@client1 ~]#mkdir /lustre

[root@client1 ~]#mount -t lustre mds1:mds2:/testfs /lustre

[root@client2 ~]#mkdir /lustre

[root@client2 ~]#mount -t lustre mds1:mds2:/testfs /lustre

四、测试

1.在 client1 的/lustre 里建立任何文件或者删除任何文件在 client2 的/lustre 去看变化

2.正常关闭 mds1 继续在两个 client 端测试文件变化情况--这个过程要等待一段时间.（很长我的虚拟机等了 5 分钟。官方推荐用 HEARTBEAT 做 IPFAIL 的 HA）这个时候要是去动客户端的话。终端要卡死。。-9 都杀不死。观察 mds2 上的日志。（我直接 DOWN 掉电源或者网卡好象是不行的。这个需要 HA 的软件支持。正常关闭 MDS1 没问题）

[root@mds1 ~]#umount /mnt/mdt

[root@client1 ~]netstst -a |more

注意观察你的连接情况。应该没有 mds2 的。切换完成后再来一次。应该能看到了连接上 mds2 了。

3.不启动 md1 继续正常关闭 oss1 oss3 基础测试变化(切换时间照样是很长)

[root@oss1 ~]#umount /mnt/ost

[root@oss3 ~]#umount /mnt/ost

4.打完手工剩余的想怎么测试就怎么测试了，开机自动挂载就自己编辑 fstab 文件了。（failover 还是介意最好用 heartbeat 一类的 software 来控制）

E-mail:qq174375@gmail.com