**手动安装手册**

目录

[1.说明 1](#_Toc464755629)

[1.1 系统环境 1](#_Toc464755630)

[1.2 Luspinf 1](#_Toc464755631)

[1.3 目标 1](#_Toc464755632)

[1.4 其它 2](#_Toc464755633)

[1.5说明 2](#_Toc464755634)

[2.主要编译、安装步骤 3](#_Toc464755635)

[2.1 Luspinf安装 3](#_Toc464755636)

[2.2 安装依赖 3](#_Toc464755637)

[2.3 安装Lustre Source 6](#_Toc464755638)

[2.4 安装Kernel Source 7](#_Toc464755639)

[2.5 编译内核 15](#_Toc464755640)

[2.6 安装Lustre服务器内核 20](#_Toc464755641)

[2.7 安装Lustre客户端内核 22](#_Toc464755642)

[2.7.1 直接安装 22](#_Toc464755643)

[2.7.2 编译安装 22](#_Toc464755644)

[2.7.3 其他安装方法 24](#_Toc464755645)

[2.8 Lustre 配置 24](#_Toc464755646)

[3. 测试 29](#_Toc464755647)

[3.1配置 29](#_Toc464755648)

[3.2 IOZone测试 31](#_Toc464755649)

[3.3 fio测试 36](#_Toc464755650)

[4.结果 39](#_Toc464755651)

[4.1 IOZone测试 39](#_Toc464755652)

[4.2 fio测试 44](#_Toc464755653)

[5.附录 45](#_Toc464755654)

[5.1 makemuconfig配置参考 45](#_Toc464755655)

[5.2 常见错误及处理方法 52](#_Toc464755656)

# 1.说明

## 1.1 系统环境

|  |  |
| --- | --- |
| 项目 | 详情 |
| 主操作系统 | Microsoft Windows 10企业版64位[版本 10.0.10586] |
| 虚拟软件 | VMware® Workstation 12 Pro 12.1.0 build-3272444 |
| 虚拟操作系统 | CentOS 6.4 [内核2.6.32-358.el6.x86\_64] |

表 1-1 系统环境

## 1.2 Luspinf

|  |  |
| --- | --- |
| Lustre | lustre-2.8.0 [Linux内核要求：linux-2.6.32-504.8.1.el6] |
| Split I/O Scheduler | 针对Linux3.2.51 |

表 1-2 Luspinf

## 1.3 目标

对Linux内核、Lustre以及Split做适当的修改，将Lustre和Split结合起来在CentOS 6.4[内核linux-2.6.32-504.8.1.el6]上运行，利用Split对Lustre做出相应的改进，并测试改进的效果。

## 1.4 其它

Lustre主页：http://lustre.org/

Lusre下载地址：<https://downloads.hpdd.intel.com/public/lustre/>

Split I/O Scheduler主页：<http://research.cs.wisc.edu/adsl/Software/split/>

Iozone主页：<http://www.iozone.org/>

## 1.5说明

1. 其它版本的Lustre以及内核相关操作可以借鉴本手册的相关步骤；
2. 现有自动化编译、部署、测试的工具可用，但在不同的配置下可能出现一些错误，可以借助本手册进行排查；另外，本手册附有常见错误以及借鉴方案供参考。

# 2.主要编译、安装步骤

## 2.1 Luspinf安装

Lustre是一个比较复杂的分布式文件系统，当前（截至到2016年3月1日）最新的版本为Lustre2.8

## 2.2 安装依赖

1. 以root身份登录，关闭SELinux服务，关闭防火墙

|  |
| --- |
| 关闭SELinux服务   1. 永久方法 – 需要重启服务器   修改vi /etc/selinux/config文件中设置SELINUX=disabled ，然后重启服务器。   1. 临时方法 – 设置系统参数   使用命令setenforce 0  附：  setenforce 1 设置SELinux 成为enforcing模式  setenforce 0 设置SELinux 成为permissive模式  getenforce ，这个命令可以查看到selinux的状态，还有一个命令也可以查看出selinux的状态：sestatus -v，假设selinux是正在运行的，我们可以使用setenforce 命令设置临时关闭，不用重启生效。 |
| 关闭防火墙   1. 重启后永久性生效：   开启：chkconfig iptables on  关闭：chkconfig iptables off   1. 即时生效，重启后失效：   开启：service iptables start  关闭：service iptables stop |

1. 安装4个链接库

yum install -y gcc-c++ && yum install -y libXi && yum install -y libXi-devel && yum install -y ncurses-devel （使用make menuconfig必须）

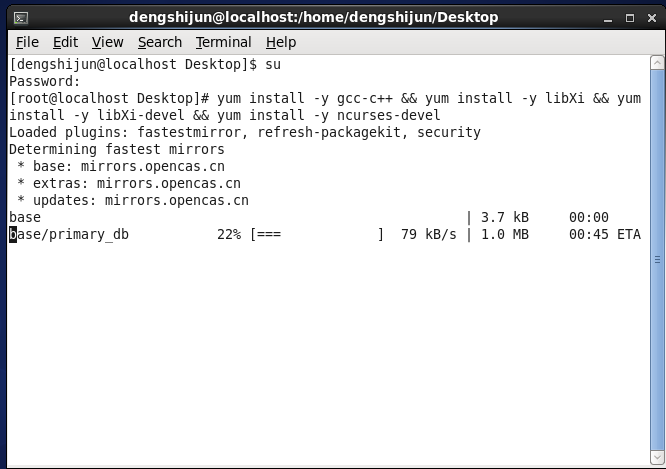


图 2-1

1. 安装开发工具

# yum -y groupinstall "Development Tools"

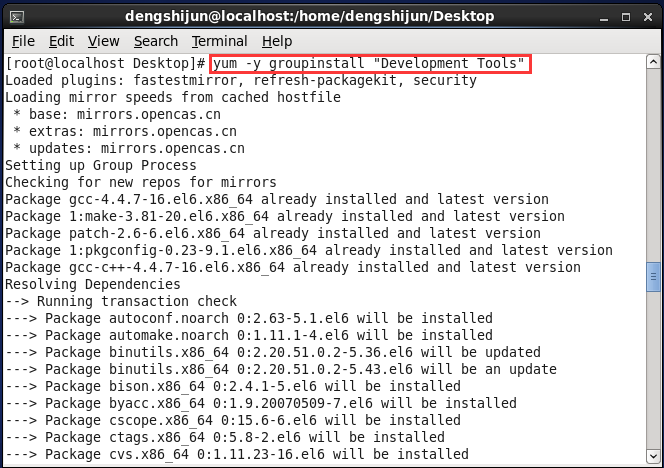


图 2-2

1. 安装其它依赖项

yum -y install xmlto && yum -y install asciidoc && yum -y install elfutils-libelf-devel && yum -y install zlib-devel && yum -y install binutils-devel && yum -y install newt-devel && yum -y install python-devel && yum -y install hmaccalc && yum -y install perl-ExtUtils-Embed && yum -y install python-docutils && yum -y install elfutils-devel && yum -y install audit-libs-devel && yum -y install libselinux-devel

1. 安装EPEL（可选）

wget http://download.fedoraproject.org/pub/epel/6/x86\_64/epel-release-6-8.noarch.rpm

rpm -ivh ./epel-release-6-8.noarch.rpm

1. 安装quit，# yum -y install quilt

注意：若此步不能成功，请进行以下操作

wget http://mirror.centos.org/centos/6/os/x86\_64/Packages/newt-devel-0.52.11-3.el6.x86\_64.rpm

wget http://mirror.centos.org/centos/6/os/x86\_64/Packages/slang-devel-2.2.1-1.el6.x86\_64.rpm

wget http://mirror.centos.org/centos/6/os/i386/Packages/asciidoc-8.4.5-4.1.el6.noarch.rpm

yum --nogpgcheck localinstall ./newt-devel-0.52.11-3.el6.x86\_64.rpm ./slang-devel-2.2.1-1.el6.x86\_64.rpm ./asciidoc-8.4.5-4.1.el6.noarch.rpm

## 2.3 安装Lustre Source

1. 在home目录下创建build目录，即/home/build：

# useradd -m build

1. 切换至用户态并进入home目录：

# su build

$ cd $HOME

1. 下载MASTER

$ git clone git://git.hpdd.intel.com/fs/lustre-release.git

$ cd lustre-release

$ sh ./autogen.sh #（解决依赖问题）

## 2.4 安装Kernel Source

1. 内核代码

$ cd $HOME

$ mkdir -p kernel/rpmbuild/{BUILD,RPMS,SOURCES,SPECS,SRPMS}

$ cd kernel

$ echo '%\_topdir %(echo $HOME)/kernel/rpmbuild'> ~/.rpmmacros

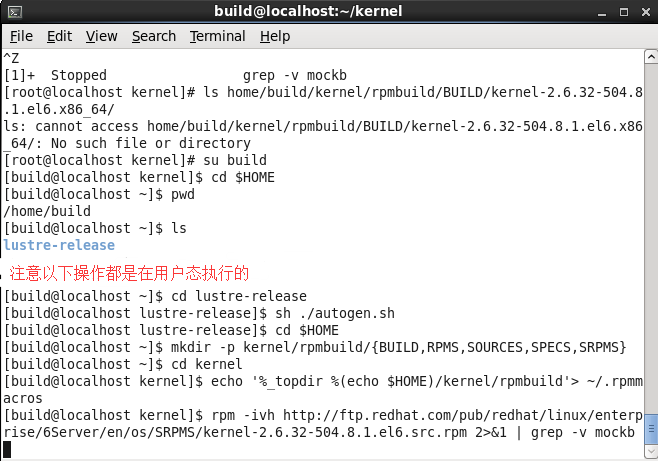


图 2-3

1. 安装

rpm -ivh http://ftp.redhat.com/pub/redhat/linux/enterprise/6Server/en/os/SRPMS/kernel-2.6.32-504.8.1.el6.src.rpm 2>&1 | grep -v mockb

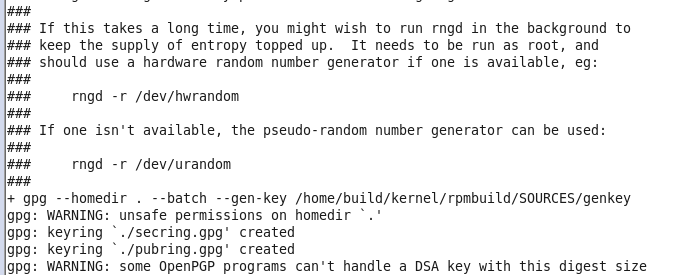
这一步操作可能需要等待一段时间

1. 进行以下操作

$ cd ~/kernel/rpmbuild

$ rpmbuild -bp --target=`uname -m` ./SPECS/kernel.spec

注：可能在一下界面停留较长的时间，请耐心等待。



命令结束后看到以下界面，表示成功：

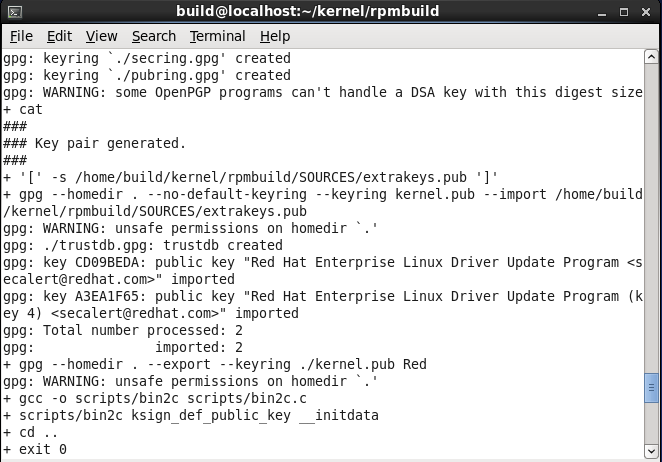


图 2-4

现在应该可以看到kernel source驻留的位置

$ ls /home/build/kernel/rpmbuild/BUILD/kernel-2.6.32-504.8.1.el6/linux-2.6.32-504.8.1.el6.x86\_64/

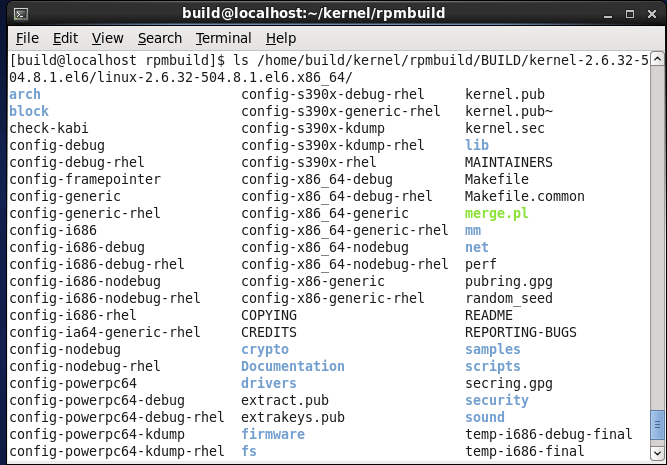


图 2-5

1. 进入内核代码目录：

$ cd ~/kernel/rpmbuild/BUILD/kernel-2.6.32-504.8.1.el6/linux-2.6.32-504.8.1.el6.x86\_64/

vi Makefile修改EXTRAVERSION = -504.8.1.el6\_lustre.x86\_64

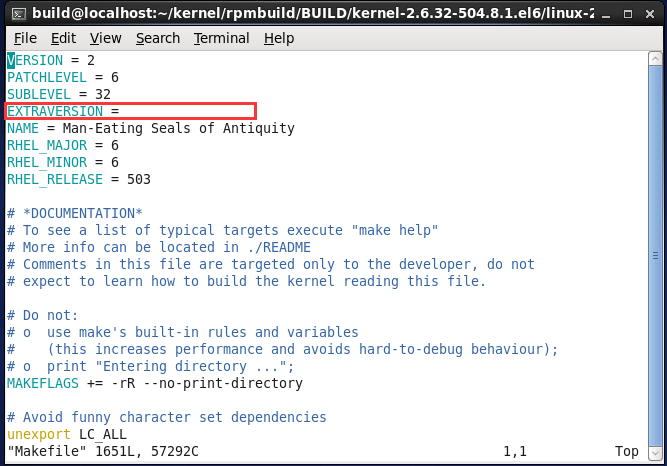


图 2-5

1. 覆盖.config

$ cp ~/lustre-release/lustre/kernel\_patches/kernel\_configs/kernel-2.6.32-2.6-rhel6-x86\_64.config ./.config

1. 打补丁

$ ln -s /home/build/lustre-release/lustre/kernel\_patches/series/2.6-rhel6.series series

$ ln -s /home/build/lustre-release/lustre/kernel\_patches/patches patches

Lustre补丁：$ for PATCH in $(cat series); do patch -p1 < patches/$PATCH; done

Split补丁：$ cd .. && patch -p1 < split\_linux2.6V20160308A.patch（补丁先要复制到当前位置，补丁完毕后再cd ~/kernel/rpmbuild/BUILD/kernel-2.6.32-504.8.1.el6/linux-2.6.32-504.8.1.el6.x86\_64/）

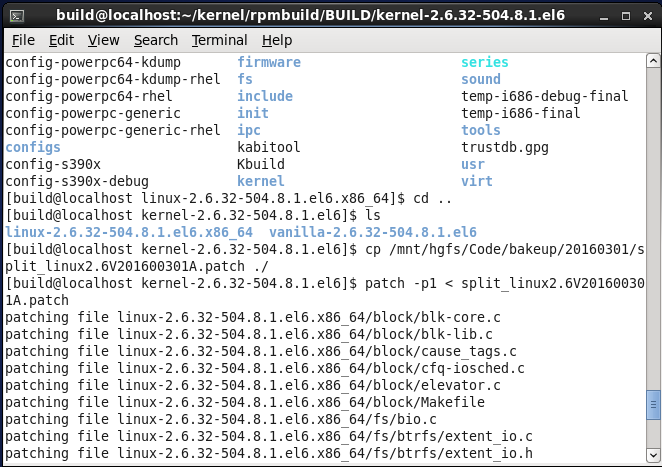


图 2-6

## 2.5 编译内核

1. 编译

$cd ~/kernel/rpmbuild/BUILD/kernel-2.6.32-504.8.1.el6/linux-2.6.32-504.8.1.el6.x86\_64/

$ make menuconfig//见附录

$ make include/asm

$ make include/linux/version.h

$ make SUBDIRS=scripts

$ make include/linux/utsrelease.h

$ make rpm （要很久）

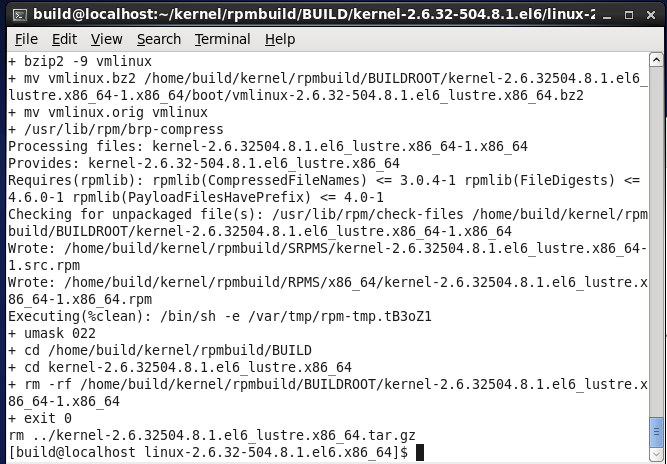


图 2-7

该过程结束后可在/home/build/kernel/rpmbuild/RPMS/x86\_64看到有kernel-2.6.32504.8.1.el6\_lustre.x86\_64-1.x86\_64.rpm文件生成

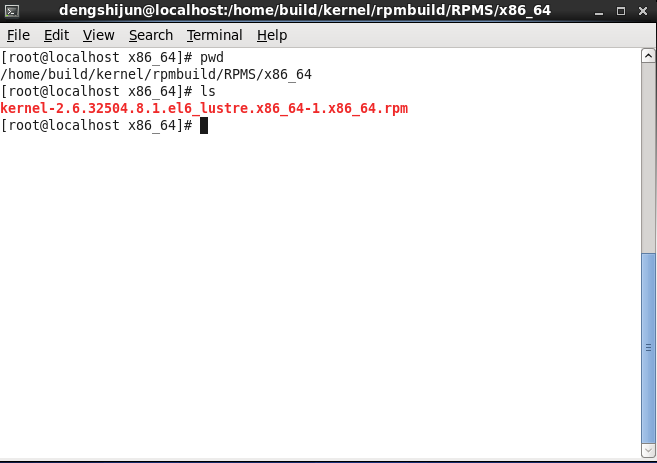


图 2-8

1. $ cd ~/lustre-release/

$ ./configure --with-linux=/home/build/kernel/rpmbuild/BUILD/kernel-2.6.32504.8.1.el6\_lustre.x86\_64

1. 生成rpm文件：$ make rpms ，这一步操作需要一定时间，操作成功，结果如下所示



图 2-9

1. 查看生成的文件

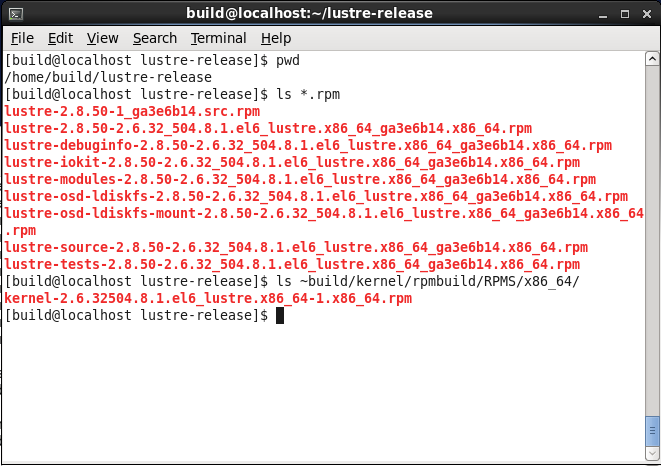


图 2-10

## 2.6 安装Lustre服务器内核

1. 以root登陆。请注意，接下来的操作是在内核态。
2. 安装内核文件# rpm -ivh ~build/kernel/rpmbuild/RPMS/x86\_64/kernel-2.6.32504.8.1.el6\_lustre.x86\_64-1.x86\_64.rpm
3. 用dracut:创建initrd

/sbin/new-kernel-pkg --package kernel --mkinitrd --dracut --depmod --install 2.6.32-504.8.1.el6\_lustre.x86\_64

1. 重启：reboot，选择新内核启动
2. rpm -ivh libcom\_err-1.42.13.wc4-7.el6.x86\_64.rpm -f

下载地址：<http://downloads.whamcloud.com/public/e2fsprogs/1.42.7.wc1/el6/RPMS/x86_64/libcom_err-1.42.7.wc1-7.el6.x86_64.rpm>

注意：如有和低版本冲突提示，先卸载低版本，例如冲突提示：

|  |
| --- |
| Preparing... ########################################### [100%]  file /lib64/libcom\_err.so.2.1 from install of libcom\_err-1.42.7.wc1-7.el6.x86\_64 conflicts with file from package libcom\_err-1.41.12-14.el6.x86\_64 |

删除低版本：rpm -e --nodeps libcom\_err-1.41.12-14.el6.x86\_64

|  |
| --- |
| 【注】如何卸载rpm包  首先通过 rpm -q <关键字> 可以查询到rpm包的名字  然后 调用 rpm -e <包的名字> 删除特定rpm包  如果遇到依赖，无法删除，使用 rpm -e --nodeps <包的名字> 不检查依赖，直接删除rpm包  如果恰好有多个包叫同样的名字，使用 rpm -e --allmatches --nodeps <包的名字> 删除所有相同名字的包， 并忽略依赖 |

1. rpm -ivh e2fsprogs-libs-1.42.13.wc4-7.el6.x86\_64.rpm -f

下载页面：<https://downloads.hpdd.intel.com/public/e2fsprogs/1.42.13.wc4/el6/RPMS/x86_64/>

注意：如有和低版本冲突提示，先卸载低版本，具体方法参照5)，示例命令：rpm -e --nodeps e2fsprogs-libs-1.41.12-14.el6.x86\_64

1. rpm -ivh e2fsprogs-1.42.13.wc4-7.el6.x86\_64.rpm -f

注意：如有和低版本冲突提示，先卸载低版本，具体方法参照5)，示例命令：rpm -e --nodeps e2fsprogs-1.41.12-14.el6.x86\_64

1. 安装Lustre RPMs
2. 进行一下安装

rpm -ivh lustre-modules-2.8.50-2.6.32\_504.8.1.el6\_lustre.x86\_64\_ga3e6b14.x86\_64.rpm --nodeps --force

rpm -ivh lustre-osd-ldiskfs-mount-2.8.50-2.6.32\_504.8.1.el6\_lustre.x86\_64\_ga3e6b14.x86\_64.rpm

rpm -ivh lustre-osd-ldiskfs-2.8.50-2.6.32\_504.8.1.el6\_lustre.x86\_64\_ga3e6b14.x86\_64.rpm

rpm -ivh lustre-2.8.50-2.6.32\_504.8.1.el6\_lustre.x86\_64\_ga3e6b14.x86\_64.rpm

yum -y install sg3\_utils

rpm -ivh lustre-iokit-2.8.50-2.6.32\_504.8.1.el6\_lustre.x86\_64\_ga3e6b14.x86\_64.rpm

rpm -ivh lustre-tests-\*

1. 测试：/usr/lib64/lustre/tests/llmount.sh

## 2.7 安装Lustre客户端内核

### 2.7.1 直接安装

1. 以root登陆。请注意，接下来的操作是在内核态。
2. 安装内核文件# rpm -ivh ~build/kernel/rpmbuild/RPMS/x86\_64/kernel-2.6.32504.8.1.el6\_lustre.x86\_64-1.x86\_64.rpm
3. 用dracut:创建initrd

/sbin/new-kernel-pkg --package kernel --mkinitrd --dracut --depmod --install 2.6.32-504.8.1.el6\_lustre.x86\_64

1. 重启：reboot，选择新内核启动
2. rpm -ivh lustre-client-modules-2.7.0-2.6.32\_504.8.1.el6.x86\_64.x86\_64.rpm --nodeps --force（可能需要安装一些缺失依赖）
3. rpm -ivh lustre-client-2.4.0-2.6.32\_358.6.2.el6.x86\_64\_gd3f91c4.x86\_64.rpm -f
4. rpm -ivh lustre-modules-2.8.50-2.6.32\_504.8.1.el6\_lustre.x86\_64\_ga3e6b14.x86\_64.rpm --nodeps -f

### 2.7.2 编译安装

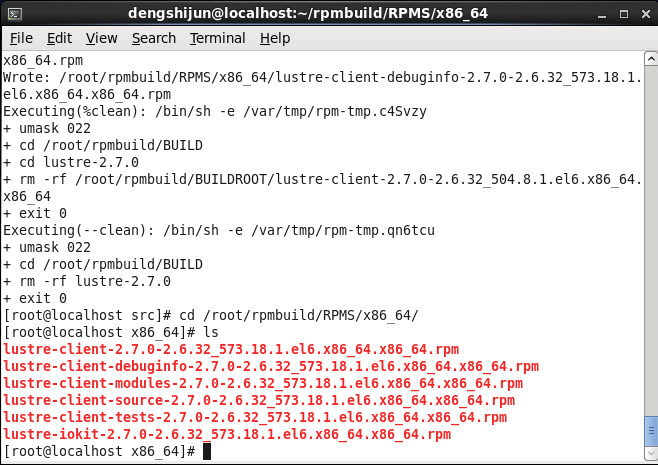
1. 下载源码包

网址：<https://downloads.hpdd.intel.com/public/lustre/lustre-2.7.0/el6/client/SRPMS/>

1. yum -y install kernel-devel rpm-build make
2. yum -y install libtool libselinux-devel
3. rpmbuild --define 'kversion 2.6.32-358.el6.x86\_64 ' --define 'kdir /usr/src/kernels/2.6.32-358.el6.x86\_64/ ' --rebuild --without servers lustre-client-2.7.0-2.6.32\_504.8.1.el6.x86\_64.src.rpm

（版本号视具体情况而定，看下/usr/src/kernels下的内核）

1. cd /root/rpmbuild/RPMS/x86\_64/中可以看到所有生成的rpm文件



1. rpm -ivh lustre-client-modules-2.7.0-2.6.32\_358.el6.x86\_64.x86\_64.rpm
2. rpm -ivh lustre-client-2.7.0-2.6.32\_358.el6.x86\_64.x86\_64.rpm
3. rpm -ivh lustre-modules-2.8.50-2.6.32\_504.8.1.el6\_lustre.x86\_64\_ga3e6b14.x86\_64.rpm --nodeps -f

### 2.7.3 其他安装方法

1. 方法一：<http://www.spheniscus.brennik.fr/doku.php/system/linux_cluster/pfs_setup>
2. 方法二：http://blog.csdn.net/wjciayf/article/details/41643397

## 2.8 Lustre 配置

lustre 的配置也是一个繁琐的过程。Lustre 是基于内核的分布式文件系统，而不是像其他一些用户态的分布式文件系统那样直接建立在 ext3 或者是 ext4 之上。Lustre 需要对磁盘进行格式化，并且在格式化的过程中进行参数配置。

|  |  |  |
| --- | --- | --- |
| IP | 角色 | 备注 |
| 192.168.150.195 | client |  |
| 192.168.150.196 | server | 有一个空余磁盘 sdb |
| 192.168.150.198 | server | 有一个空余磁盘 sdb |
| 192.168.150.199 | server | 有一个空余磁盘 sdb |

表 2-1 lustre结点配置说明

假设共有四台机器，IP分别为 192.168.150.195/192.168.150.196/192.168.150.198/192.168.150.199 ，server各有一个空余磁盘 sda4（具体参照fdisk -l命令具体设置），192.168.150.195 为客户端。在角色为server的机器中安装服务端 rpm 包，client中安装客户端 rpm 包，具体安装过程参考如上。

1. **MDS 配置**

|  |  |  |
| --- | --- | --- |
| 192.168.150.198 | server | mkfs.lustre --fsname=lustrefs --mgs --mdt --index=0 /dev/sda4 |

表 2-2 lustre结点配置说明

【参数说明】

Fsname： 指定创建 lustre 时的文件系统名

mgs ：指定该机器为元数据服务器，即该机器为 mds

mdt ：指定/dev/sda4 为元数据实际数据存储位置

至于 index 则指定该 mgs 的索引号，mgs 可以设置主备模式，但 mdt 需要在主备 mds 之间共享

1. **3.2 OSS 配置**

Mkfs.lustre

|  |  |  |
| --- | --- | --- |
| 192.168.150.196 | server | mkfs.lustre --fsname=lustrefs --mgsnode=192.168.150.198@tcp --ost --index=0 /dev/sda4 |
| 192.168.150.199 | server | mkfs.lustre --fsname=lustrefs --mgsnode=192.168.150.198@tcp --ost --index=1 /dev/sda4 |

表 2-3 lustreOSS结点配置说明

【参数说明】

fsname ：指定的是创建lustre时的文件系统名

mgsnode ：指定 oss 请求元数据节点，并指明访问协议 tcp

ost： 指定/dev/sda4 为存储数据的实际位置

index:：这个参数可以看到两个节点是不一样的

整个系统的实际容量是 192.168.150.196和192.168.150.199 的/dev/sda4的容量之和

1. **3.3 系统挂载**

MDS(192.168.150.198):

图 2-11 lustre结点分布

mount -t lustre /dev/sda4 /mnt/mdt

OSS(192.168.150.196)

192.168.150.195(client)

192.168.150.196(server)

192.168.150.198(server)

192.168.150.199(server)

mount -t lustre /dev/sda4 /mnt/ost0

OSS(192.168.150.199)

mount -t lustre /dev/sda4 /mnt/ost1

Client(192.168.150.195):

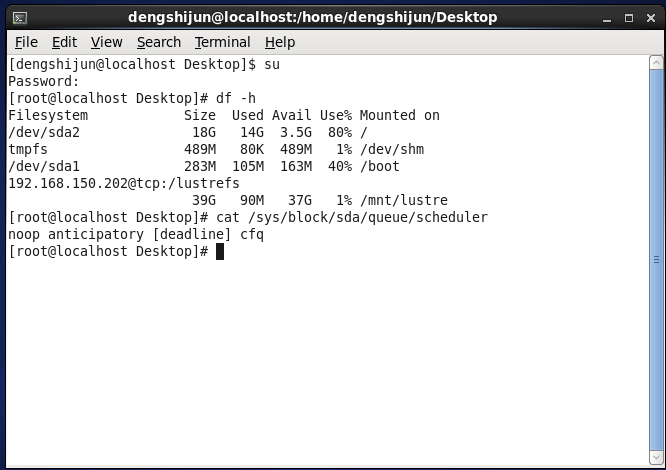
mount -t lustre 192.168.150.198@tcp:/lustrefs /mnt/lustre在客户端的/mnt/lustre 上就可以看到 192.168.150.196/192.168.150.199 的/dev/sdb 的容量此配置完毕，可以使用了。

图 2-12

# 测试

## 3.1配置

1. 以root登陆。查看系统的IO调度策略：#cat /sys/block/sda/queue/scheduler
2. 将split调度模块注册，insmod split-deadline-iosched.ko
3. 切换IO调度策略：echo new\_sysdeadline > /sys/block/sda/queue/scheduler

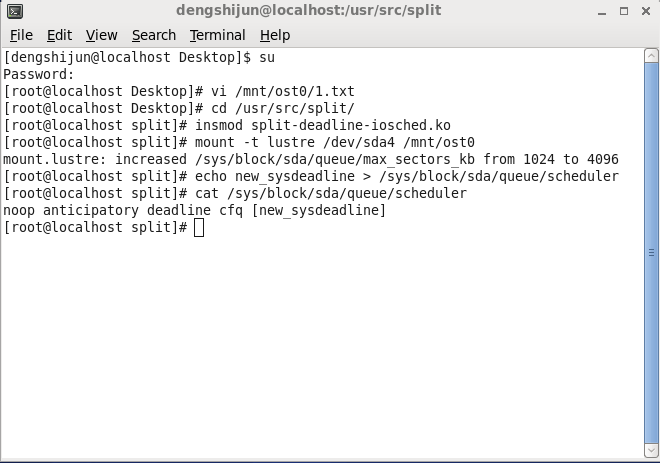
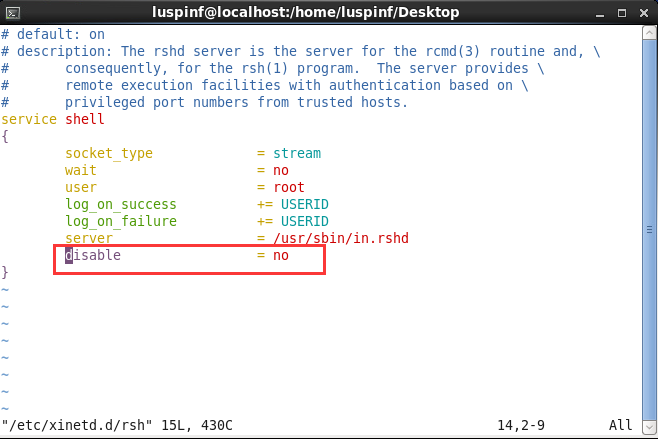


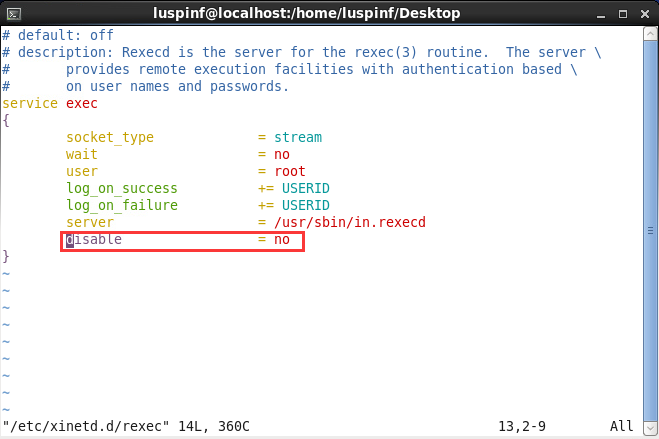
图 3-1

## 3.2 IOZone测试

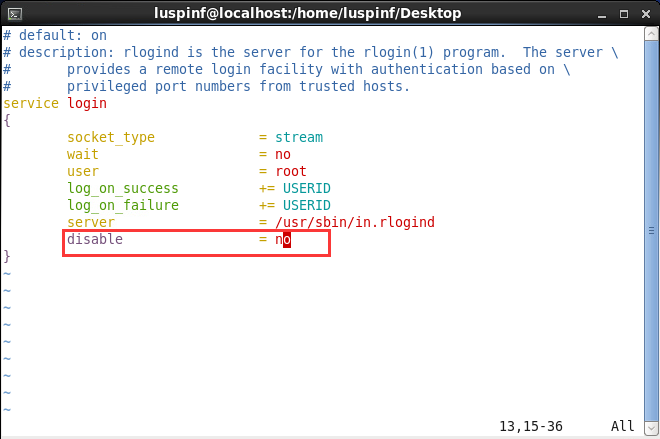
1. 下载安装iozone.
2. rsh 安装(所有需要测试节点) yum -y install rsh && yum -y install rsh-server.
3. vim /etc/xinetd.d/rsh



1. vim /etc/xinetd.d/rexec



1. vim /etc/xinetd.d/rlogin



1. echo "rsh" >> /etc/securetty && echo "rexec" >> /etc/securetty && echo "rlogin" >> /etc/securetty










13. 利用iozone测试：/opt/iozone/bin/iozone -a -s 2G -f /mnt/lustre/test/test.txt -i 0 -i 1 -i 2 -Rb ./iozone-lustreB.xls

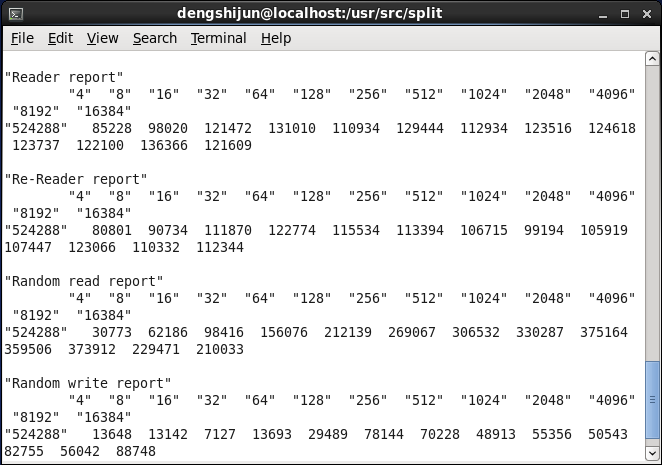
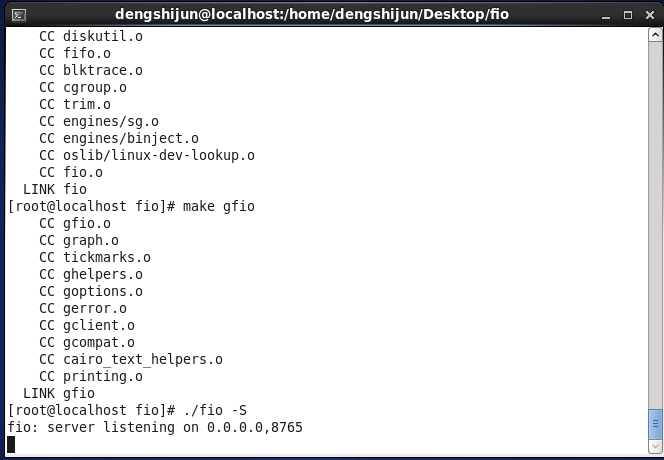


图 3-2

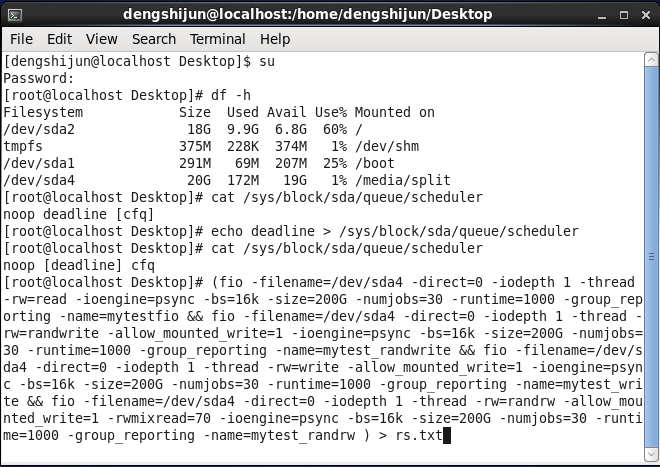
## 3.3 fio测试

1. 安装GTK+的开发文件和GLib库：yum -y install gtk2-devel && yum -y install glib2-devel
2. git clone git://git.kernel.dk/fio.git （若没有安装git，先安装git：yum -y install git）
3. 进入fio，cd fio，然后执行./configure --enable-gfio
4. make fio
5. make gfio
6. ./fio -S



1. 这样fio就编译好了，同时进入server模式。 在另外一个终端运行 gfio 就可以看到图形界面
2. 测试命令

|  |
| --- |
| fio -filename=/dev/sda4 -direct=0 -iodepth 1 -thread -rw=read -ioengine=psync -bs=16k -size=20G -numjobs=30 -runtime=1000 -group\_reporting -name=mytest\_read > rs.txt&& fio -filename=/dev/sda4 -direct=0 -iodepth 1 -thread -rw=randwrite -allow\_mounted\_write=1 -ioengine=psync -bs=16k -size=20G -numjobs=30 -runtime=1000 -group\_reporting -name=mytest\_randwrite >> rs.txt && fio -filename=/dev/sda4 -direct=0 -iodepth 1 -thread -rw=write -allow\_mounted\_write=1 -ioengine=psync -bs=16k -size=20G -numjobs=30 -runtime=1000 -group\_reporting -name=mytest\_write >> rs.txt && fio -filename=/dev/sda4 -direct=0 -iodepth 1 -thread -rw=randrw -allow\_mounted\_write=1 -rwmixread=50 -ioengine=psync -bs=16k -size=20G -numjobs=30 -runtime=1000 -group\_reporting -name=mytest\_randrw >> rs.txt |

1. 

# 4.结果

## 4.1 IOZone测试

图 4-1

图 4-2

图 4-3

图 4-4

图 4-5

图 4-6

## 4.2 fio测试

见测试报告

# 5.附录

## 5.1 makemuconfig配置参考

1. 去掉Enable loadable module support ---> [\*]Module signature verification (EXPERIMENTAL)

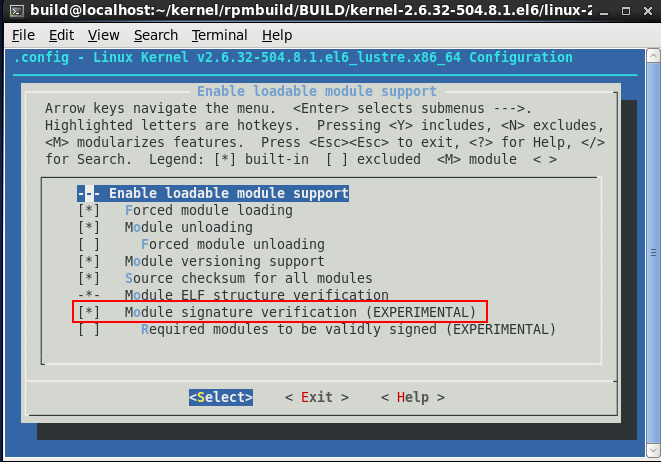


图 5-1

1. 去掉-\*- Cryptographic API ---> [ \*] In-kernel signature checker (EXPERIMENTAL)

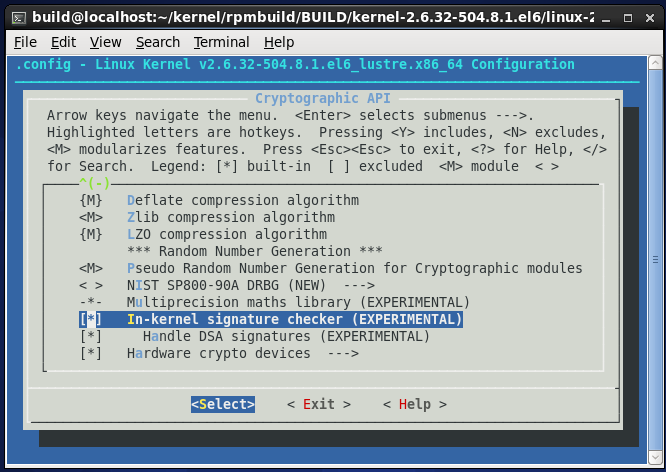


图 5-2

1. Device Drivers ─[\*] Fusion MPT device support ---> 如下配置

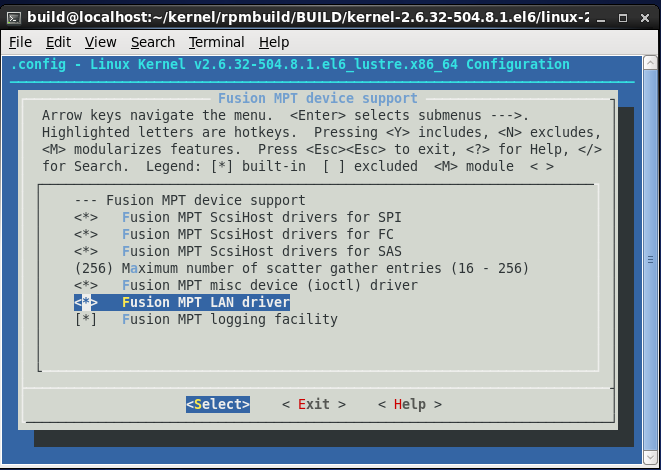


图 5-3

1. Device Drivers ─SCSI device support ---> 如下配置

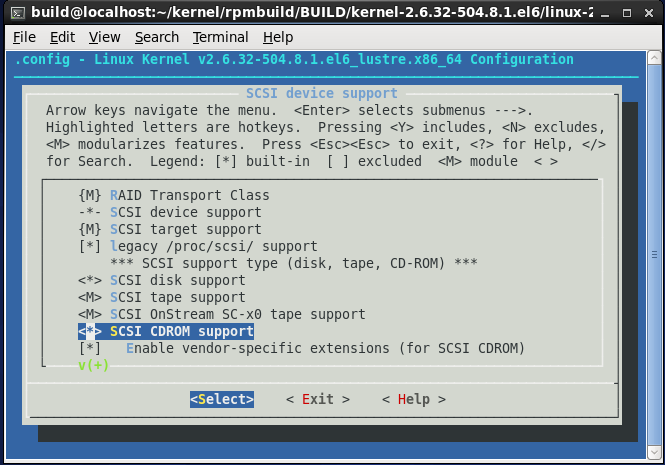


图 5-4

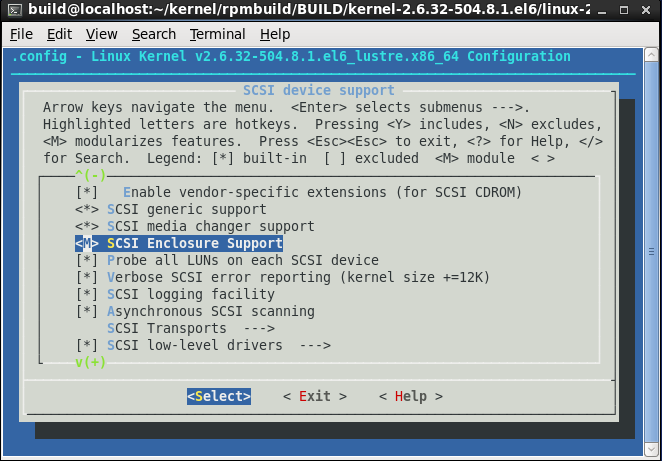


图 5-5

1. File systems ──Ext3配置

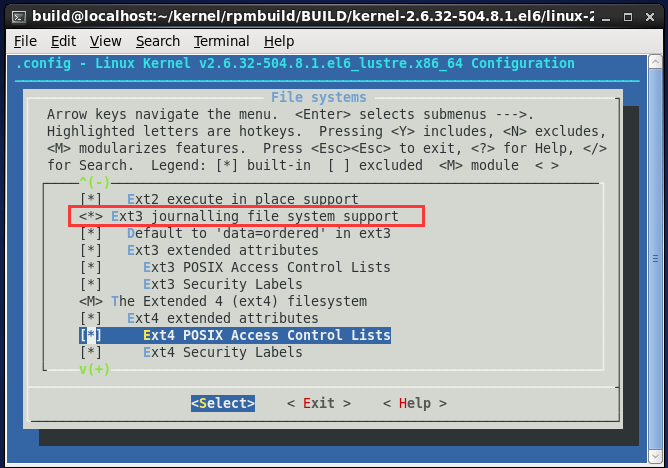


图 5-6

1. File systems ──Btrfs配置

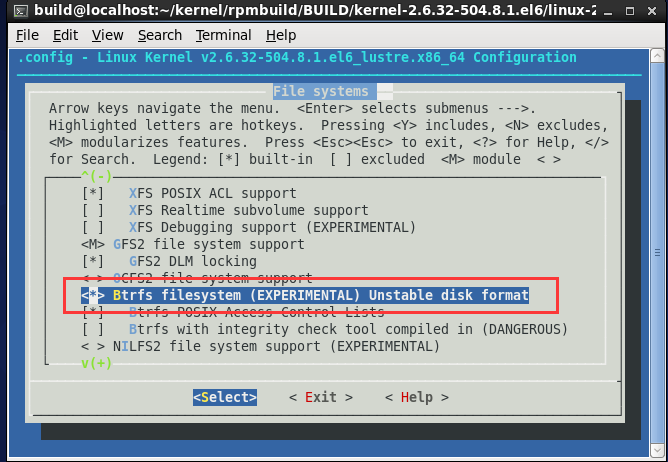


图 5-7

## 5.2 常见错误及处理方法

1. 安装依赖时Error: Cannot retrieve metalink for repository: epel. Please verify its path and try again

解决方案：

修改文件“vi /etc/yum.repos.d/epel.repo”， 将baseurl的注释取消， mirrorlist注释掉，即可。

来自 <http://blog.csdn.net/edwzhang/article/details/41251015>

或者参见 <http://blog.sina.com.cn/s/blog_4a5eb9980102v2e1.html>

1. cannot find a valid baseurl for repo:update

<https://www.centos.org/forums/viewtopic.php?f=30&t=34427>

vi /etc/yum.repos.d CentOS-Base.repo

[base]

name=CentOS-$releasever - Base

mirrorlist=http://mirrorlist.centos.org/?release=$releasever&arch=$basearch&repo=os

gpgcheck=1

gpgkey=http://mirror.centos.org/centos/RPM-GPG-KEY-centos4

enabled=1

[update]

name=CentOS-$releasever - Updates

mirrorlist=http://mirrorlist.centos.org/?release=$releasever&arch=$basearch&repo=updates

gpgcheck=1

gpgkey=http://mirror.centos.org/centos/RPM-GPG-KEY-centos4

enabled=1

[addons]

name=CentOS-$releasever - Addons

mirrorlist=http://mirrorlist.centos.org/?release=$releasever&arch=$basearch&repo=addons

gpgcheck=1

gpgkey=http://mirror.centos.org/centos/RPM-GPG-KEY-centos4

enabled=0

[extras]

name=CentOS-$releasever - Extras

mirrorlist=http://mirrorlist.centos.org/?release=$releasever&arch=$basearch&repo=extras

gpgcheck=1

gpgkey=http://mirror.centos.org/centos/RPM-GPG-KEY-centos4

enabled=0

[centosplus]

name=CentOS-$releasever - Plus

mirrorlist=http://mirrorlist.centos.org/?release=$releasever&arch=$basearch&repo=centosplus

gpgcheck=1

enabled=0

gpgkey=http://mirror.centos.org/centos/RPM-GPG-KEY-centos4

[contrib]

name=CentOS-$releasever - Contrib

mirrorlist=http://mirrorlist.centos.org/?release=$releasever&arch=$basearch&repo=contrib

gpgcheck=1

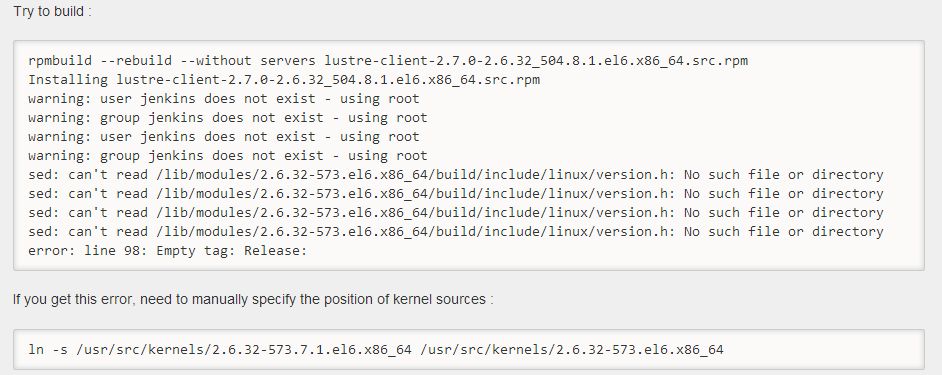
enabled=0

gpgkey=http://mirror.centos.org/centos/RPM-GPG-KEY-centos4

1. ed: can't read /lib/modules/2.6.32-504.8.1.el6\_lustre.x86\_64/build/include/linux

解决方法一：rpmbuild --define 'kversion 2.6.32-573.18.1.el6.x86\_64 ' --define 'kdir /usr/src/kernels/2.6.32-573.18.1.el6.x86\_64/ ' --rebuild --without servers lustre-client-2.7.0-2.6.32\_504.8.1.el6.x86\_64.src.rpm

解决方案二：



1. mount -t lustre /dev/sda4 /mnt/mdt

问题描述：

[root@localhost Desktop]# mount -t lustre /dev/sda4 /mnt/mdtmount.lustre: mount /dev/sda4 at /mnt/mdt failed: No such device

Are the lustre modules loaded?

Check /etc/modprobe.conf and /proc/filesystems

If I explicitly preload lustre modules with a modprobe command like "modprobe lustre" then the mount works fine.

解决方案一：

I got the following commentary in email from Andreas:

This is a problem known to me. If the obdclass module is loaded, then it will register the "lustre" filesystem type, so it will appear in /proc/filesystems and mount will not modprobe the "lustre" filesystem module.

This has been true since Lustre 1.6 or so (when "mountconf" was first added).

Two options exist:

#1 modify the mount.lustre binary to always modprobe the "lustre" module if it isn't already loaded

#2 change the "lustre" filesystem type registration to be in the "lustre" module (as most other filesystems do). That has the problem that the servers do not need the "lustre.ko" module loaded, since that is really the client VFS interface. It would help if there was a second filesystem type "lustre\_srv" or similar, that could be used to register server mountpoints, and possibly simplify the mount internals (which are convoluted because they have to do completely different things for client and server mounts).

Obviously, #1 is easier, but #2 would simplify the coe in the long term.

解决方案二：网络是否正常



[root@localhost Desktop]# yum -y groupinstall "Development Tools"

Loaded plugins: fastestmirror, refresh-packagekit, security

Loading mirror speeds from cached hostfile

\* base: centos.ustc.edu.cn

\* extras: ftp.sjtu.edu.cn

\* updates: centos.ustc.edu.cn

Setting up Group Process

Checking for new repos for mirrors

Failed to add groups file for repository: base - comps file is empty/damaged

Warning: Group development does not have any packages.

No packages in any requested group available to install or update

解决方法：

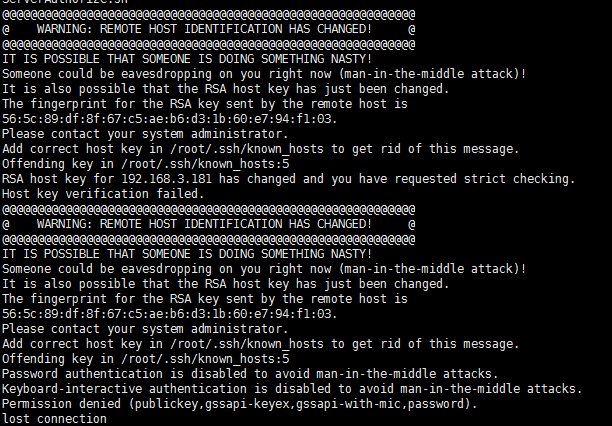
cd /etc/yum.repo.d

wget http://public-yum.oracle.com/public-yum-el5.repo

cd /etc/pki/rpm-gpg/

wget https://public-yum.oracle.com/RPM-GPG-KEY-oracle-el5

1. 配置RSA信任关系时出现如下错误，且后面远程执行命令时还是需要输入密码：



解决方法：

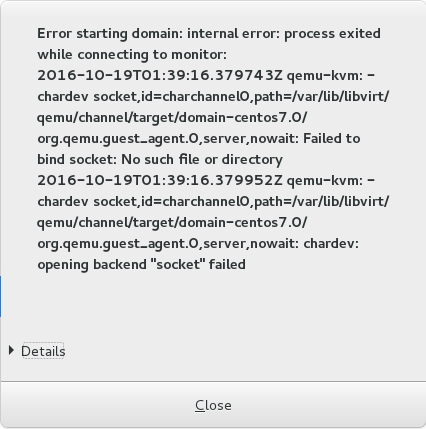
sed -i 's/SELINUX=enforcing/SELINUX=disabled/g' /etc/selinux/config;

重启；

在控制节点上执行rm -rf /root/.ssh

1. 测试过程中的问题：fio: pid=20398, err=5/file:backend.c:492, func=full resid, error=Input/output error
2. 在使用kvm虚拟机的时候克隆后的虚拟机有时候打不开，以新clone的虚拟机为/centos7.0-clone为例，打开的时候错误提示：

|  |
| --- |
| Error starting domain: internal error: process exited while connecting to monitor: 2016-10-19T01:39:16.379743Z qemu-kvm: -chardev socket,id=charchannel0,path=/var/lib/libvirt/qemu/channel/target/domain-centos7.0/org.qemu.guest\_agent.0,server,nowait: Failed to bind socket: No such file or directory  2016-10-19T01:39:16.379952Z qemu-kvm: -chardev socket,id=charchannel0,path=/var/lib/libvirt/qemu/channel/target/domain-centos7.0/org.qemu.guest\_agent.0,server,nowait: chardev: opening backend "socket" failed  Traceback (most recent call last):  File "/usr/share/virt-manager/virtManager/asyncjob.py", line 89, in cb\_wrapper  callback(asyncjob, \*args, \*\*kwargs)  File "/usr/share/virt-manager/virtManager/asyncjob.py", line 125, in tmpcb  callback(\*args, \*\*kwargs)  File "/usr/share/virt-manager/virtManager/libvirtobject.py", line 83, in newfn  ret = fn(self, \*args, \*\*kwargs)  File "/usr/share/virt-manager/virtManager/domain.py", line 1433, in startup  self.\_backend.create()  File "/usr/lib64/python2.7/site-packages/libvirt.py", line 1029, in create  if ret == -1: raise libvirtError ('virDomainCreate() failed', dom=self)  libvirtError: internal error: process exited while connecting to monitor: 2016-10-19T01:39:16.379743Z qemu-kvm: -chardev socket,id=charchannel0,path=/var/lib/libvirt/qemu/channel/target/domain-centos7.0/org.qemu.guest\_agent.0,server,nowait: Failed to bind socket: No such file or directory  2016-10-19T01:39:16.379952Z qemu-kvm: -chardev socket,id=charchannel0,path=/var/lib/libvirt/qemu/channel/target/domain-centos7.0/org.qemu.guest\_agent.0,server,nowait: chardev: opening backend "socket" failed |



解决方案：

vim /etc/libvirt/qemu/centos7.0-clone.xml



<source mode='bind' path='/var/lib/libvirt/qemu/channel/target/domain-centos7.0/org.qemu.guest\_agent.0'/>

要做修改

<source mode='bind' path='/var/lib/libvirt/qemu/channel/target/domain-centos7.0-clone/org.qemu.guest\_agent.0'/>

重启主机之后就好了

1. 安装新内核后提示ssh: connection reset by peer

解决办法：

排除错误步骤：

ssh –v：ssh -v root@10.1.101.35

参见<http://www.cnblogs.com/starof/p/4709805.html>

其他可能原因：没有安装libcom\_err-1.42.13.wc5-7.el7.x86\_64.rpm、libss-1.42.13.wc5-7.el7.x86\_64.rpm

1. 编译内核的时候根分区保留20G以上的空间比较好，否则会出现根分区满无法继续编译而莫宁奇妙的错误
2. ssh ip没有反应，无任何提示：

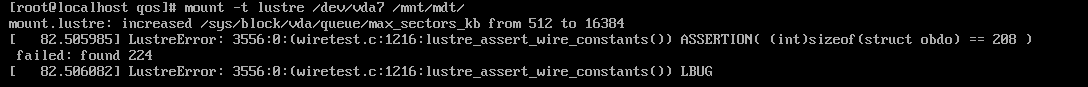
1.SSH是否安装、启动、防火墙、端口占用

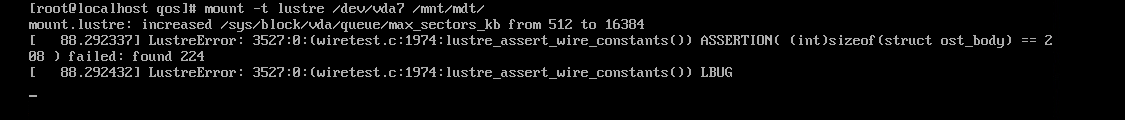
2.子网掩码设置错误

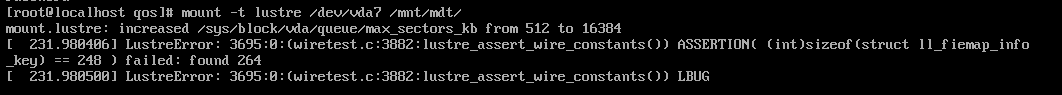
1. mds挂载lustre死机重启，原因可能不尽相同，要从日志方面解决，挂载死机的时候显示的提示（要很快截屏，否则崩溃后看不到该提示）

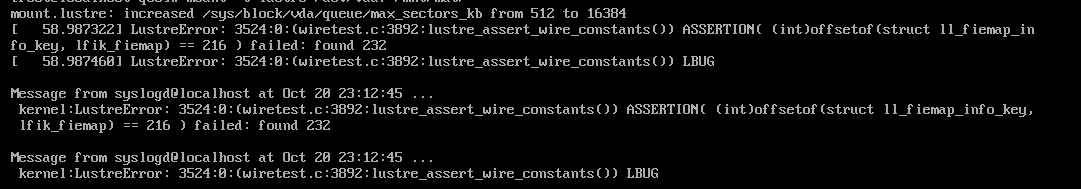
例如已经遇到过的问题，日志:/var/log/messages libcfs: module verification failed: signature and/or required key missing - tainting kernel 是.config没配好或者lustre自带的没有使用。

又例如，如图所示：（显示的提示要很快截屏，否则崩溃后看不到该提示）









1. 编译client rpm时注意加上--disable-server ，否则可能得到的是server端的
2. client一写文件就死机