**Linux下DRBD配置**

一、drbd 工作原理

DRBD是一种块设备,可以被用于高可用(HA)之中.它类似于一个网络RAID-1功能.当你将数据写入本地 文件系统时,数据还将会被发送到网络中另一台主机上.以相同的形式记录在一个文件系统中。 本地(主节点)与远程主机(备节点)的数据可以保证实时同步.当本地系统出现故障时,远程主机上还会 保留有一份相同的数据,可以继续使用.在高可用(HA)中使用DRBD功能,可以代替使用一个共享盘阵.因为数据同时存在于本地主机和远程主机上,切换时,远程主机只要使用它上面的那份备份数据,就可以继续进行服务了。

二、环境配置：

1、操作系统：radhat 5.4

2、主服务器：

主机名：master

IP：192.168.2.124

镜像的硬盘：/dev/hdb

3、  备份服务器

主机名：slave

IP：192.168.2.125

镜像的硬盘：/dev/hdb

三、安装drbd

1、下载源代码：<http://oss.linbit.com/drbd/>。此次安装的版本是drbd-8.3.5

2、把下载好的drbd-8.3.5.tar.gz解压

[root@master ] # tar –zxvf drbd-8.3.5.tar.gz

[root@master ] # cd drbd-8.3.5

[root@master ] # make

[root@master ] #  make install

[root@master ]# modprobe drbd            @加载安装drbd模块

[root@master ]# lsmod |grep drbd

drbd         271736  0

@通过lsmod检查是否已经成功，如果有类似内容输出，则表示drbd安装成功了

[root@master ]# mknod /dev/drbd0 b 147 0    @创建硬件设备drbd

四、配置drbd

DRBD运行时,会读取一个配置文件/etc/drbd.conf.这个文件里描述了DRBD设备与硬盘分区的映射关系,数据一旦写入磁盘并发送到网络中就认为完成了写入操作。

1、drbd.conf的配置参数说明

Protocol

Protocol  A         @数据一旦写入磁盘并发送到网络中就认为完成了写入操作

Protocol  B         @收到接收确认就认为完成了写入操作。

Protocol  C         @收到写入确认就认为完成了写入操作。

2、global

global { usage-count yes; }         @是否参加DRBD使用者统计，默认是yes

3、common

common { syncer { rate 1M; } }

@设置主备节点同步时的网络速率最大值,单位是字节.

4、resource

一个DRBD设备(即:/dev/drbdX),叫做一个"资源"。里面包含一个DRBD设备的主备节点的的ip信息，底层存储设备名称，设备大小，meta信息存放方式，drbd对外提供的设备名等等。

resource r0 {

protocol C;         @使用协议C.表示收到远程主机的写入确认后,则认为写入完成.

net {

     cram-hmac-alg sha1;            @设置主备机之间通信使用的信息算法.

     shared-secret "FooFunFactory";

   }

@每个主机的说明以"on"开头,后面是主机名.在后面的{}中为这个主机的配置.

on master {

    device    /dev/drbd0;

    disk      /dev/ hdb;

    address   192.168.2.124:7898;    @设置DRBD的监听端口,用于与另一台主机通信

    meta-disk  internal;

   }

on slave{

    device    /dev/drbd0;

    disk      /dev/ hdb;

    address   192.168.2.124:7898;

     meta-disk  internal;

    }

 }

5、以下为测试时的完整配置

global {

  usage-count yes;

}

common {

  protocol C;

}

resource r0 {

  on master {

    device    /dev/drbd0;

    disk      /dev/hdb;

    address   192.168.2.124:7789;

    meta-disk  internal;

  }

  on slave {

    device    /dev/drbd0;

    disk      /dev/hdb;

    address   192.168.2.125:7789;

    meta-disk internal;

  }

}

6、双机两个节点的配置文件是一模一样的。DRBD源码目录下有个样本，里面说得比较详细。

五、启动drbd

1、创建matadata

在启动DRBD之前,需要分别在两台主机的hdb分区上,创建供DRBD记录信息的数据块.分别在两台主机上执行:

[root@master ]#drbdadm create-md r0

[root@slave ]#drbdadm create-md r0

备注：

1) “r0”是在drbd.conf里定义的资源名称.

2) 当执行命令”drbdadm create-md r0”时，出现以下错误信息。

*Device size would be truncated, which*

*would corrupt data and result in*

*'access beyond end of device' errors.*

*You need to either*

*\* use external meta data (recommended)*

*\* shrink that filesystem first*

*\* zero out the device (destroy the filesystem)*

*Operation refused.*

*Command 'drbdmeta 0 v08 /dev/xvdb internal create-md' terminated with exit code 40*

*drbdadm create-md r0: exited with code 40*

解决办法：初始化磁盘文件格式, dd if=/dev/zero bs=1M count=1 of=/dev/sdXYZ; sync

[root@master ]#  dd if=/dev/zero bs=1M count=1 of=/dev/hda3; sync

2.启动DRBD，分别在两台主机上执行

[root@master ]# /etc/init.d/drbd start

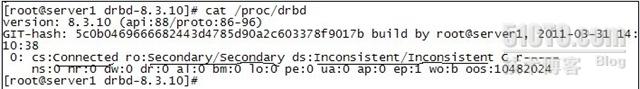
[root@slave ]# /etc/init.d/drbd start

备注：

如果在启动过程中找不到node时，查看是/dev/hda3是否是处于挂载状态，如果是,用umount命令取消挂载

3. 查看DRBD的状态，分别在两台主机上执行

[root@slave drbd]# cat /proc/drbd

[](http://img1.51cto.com/attachment/201206/151842211.jpg)

对输出的含义解释如下：

ro表示角色信息，第一次启动drbd时，两个drbd节点默认都处于Secondary状态，

ds是磁盘状态信息，“Inconsistent/Inconsisten”，即为“不一致/不一致”状态，表示两个节点的磁盘数据处于不一致状态。

Ns表示网络发送的数据包信息。

Dw是磁盘写信息

Dr是磁盘读信息

4、设置主节点

由于默认没有主次节点之分，因而需要设置两个主机的主次节点，选择需要设置为主节点的主机，然后执行如下命令：

[root@master]#drbdsetup /dev/drbd0 primary –o

或者执行下面命令也是可以的

[root@ master]#drbdadm -- --overwrite-data-of-peer primary all

第一次执行完此命令后，在后面如果需要设置哪个是主节点时，就可以使用另外一个命令：

[root@ master]#/sbin/drbdadm primary r0或者/sbin/drbdadm primary all

再次查看drbd状态，

[](http://img1.51cto.com/attachment/201206/151918371.jpg)

可以看到现在状态已经是primary/secondary，而且磁盘信息是upTodat/Inconsistent（实时/不一致）,而且已经开始同步两台机器对应磁盘的数据

过了一会，我们再次查看drbd状态，如下：

[](http://img1.51cto.com/attachment/201206/151943158.jpg)

此时的状态信息就是primary/secondary，磁盘信息是UpToDate/UpToDate，说明已经同步完成了。

六、测试同步

把主机上的DRBD设备挂载到一个目录上进行使用。备机的DRBD设备无法被挂载,因为它是用来接收主机数据的,由DRBD负责操作。

主备节点切换有两种方式，分别是停止drbd服务切换和正常切换，依次介绍：  
1、停止drbd服务切换  
关闭主节点服务，此时挂载的drbd分区就自动在主节点卸载了，然后在备用节点执行切换命令：  
[root@drbd2 ~]#drbdadm primary all  
此时会报错：  
2: State change failed: (-7) Refusing to be Primary while peer is not outdated  
Command 'drbdsetup 2 primary' terminated with exit code 11  
因此，必须在备用节点执行如下命令：  
[root@drbd2 ~]#drbdsetup /dev/drbd0 primary –o  
或者  
[root@drbd2~]#drbdadm -- --overwrite-data-of-peer primary all  
此时就可以正常切换了。  
当在备用节点执行切换到主节点命令后，原来的主用节点自动变为备用节点。无需在主用节点再次执行切换到备用节点的命令。  
2、正常切换  
在主节点卸载磁盘分区，然后执行  
[root@drbd1 ~]#drbdadm secondary all  
如果不执行这个命令，直接在备用节点执行切换到主节点的命令，会报错：  
2: State change failed: (-1) Multiple primaries not allowed by config  
Command 'drbdsetup 2 primary' terminated with exit code 11  
接着，在备用节点执行  
[root@drbd2 ~]#drbdadm primary all  
最后在备用节点挂载磁盘分区即可：  
[root@drbd2 ~]#mount /dev/drbd0 /mnt

3、注意点

1）mount drbd设备以前必须把设备切换到primary状态。

2）两个节点中，同一时刻只能有一台处于primary状态，另一台处于secondary状态。

3）处于secondary状态的服务器上不能加载drbd设备。

4）主备服务器同步的两个分区大小最好相同,这样不至于浪费磁盘空间,因为drbd磁盘镜像相当于网络raid 1。

查阅相关文档：

<http://www.cnblogs.com/feisky/archive/2011/12/25/2310346.html>

<http://lucklong.blog.51cto.com/23691/591804>

<http://blog.sina.com.cn/s/blog_68a062130100jk1k.html>

<http://www.ixdba.net/a/os/linux/2010/1011/795.html>

**文件同步系统-DRBD**

 DRBD 是由内核模块和相关脚本而构成，用以构建高可用性的集群。其实现方式是通过网络来镜像整个设备。您可以把它看作是一种网络RAID。

DRBD负责接收数据，把数据写到本地磁盘，然后发送给另一个主机。另一个主机再将数据存到自己的磁盘中。其他所需的组件有集群成员服    务，如TurboHA 或 心跳连接，以及一些能在块设备上运行的应用程序。例如：裸I/O、文件系统及fsck、具有恢复能力的数据库。

系统环境:rhel6 x86\_64 selinux and iptables disabled

软件下载:http://oss.linbit.com/drbd

1. 安装DRBD

＃ yum -y install gcc flex rpm-build kernel-devel -y   # 解决依赖性

＃ rpmbuild ~   # 在家目录生成rpmbuild编译所需环境

＃ tar xf drbd-8.4.0.tar.gz

# cd drbd-8.4.0

# ./configure

# make rpm    # 编译drbd

# make km-rpm # 编译drbd内核模块

# cd ~/rpmbuild/RPMS/X86\_64

# rpm -ivh \*

Note:　从2.6.33开始drbd已经包含在内核中

Note:　拷贝生成的rpm包到另一主机，并安装软件包

2. 配置DRBD

# cp /usr/share/doc/drbd-utils-8.4.0/drbd.conf.example /etc/drbd.d/example.res

# vi /etc/drbd.d/example.res

resource example {

options {

on-no-data-accessible suspend-io;

}

net {

cram-hmac-alg "sha1";

shared-secret "secret\_string";

}

# The disk section is possible on resource level and in each

# volume section

disk {

# If you have a resonable RAID controller

# with non volatile write cache (BBWC, flash)

disk-flushes no;

disk-barrier no;

md-flushes no;

}

# volume sections on resource level, are inherited to all node

# sections. Place it here if the backing devices have the same

# device names on all your nodes.

on server23.example.com {

address 192.168.0.123:7780;

volume 0 {

      device minor 0;

      disk /dev/vdb1;

      meta-disk internal;

}

}

on vserver2.example.com {

address 192.168.0.223:7780;

volume 0 {

      device minor 0;

      disk /dev/vdb1;

      meta-disk internal;

}

}

}

Note: 两台机器的配置一样，直接复制

3. 启动和测试

在两台主机上分别执行以下命令

#　drbdadm create-md example

# /etc/init.d/drbd start

将其中一台机器设置为primary节点， 并同步数据

# drbdsetup /dev/drbd0 primary --force

在两台主机上查看同步状态

# watch cat /proc/drbd

version: 8.4.0 (api:1/proto:86-100)

GIT-hash: 28753f559ab51b549d16bcf487fe625d5919c49c build by root@server36.example.com,

2011-08-14 09:44:01

0: cs:SyncSource ro:Primary/Secondary ds:UpToDate/Inconsistent C r-----

ns:126192 nr:0 dw:0 dr:126856 al:0 bm:7 lo:0 pe:145 ua:0 ap:0 ep:1 wo:d oos:922896

[=>..................] sync'ed: 12.2% (922896/1048508)K

finish: 0:02:41 speed: 5,692 (4,484) K/sec

数据同步结束后创建文件系统

# mkfs.ext4 /dev/drbd0

挂载文件系统

# mount /dev/drbd0 /var/www/html

存放数据

# cp -r /etc/\* /var/www/html

卸载文件系统

# umount /dev/drbd0

将这台机器设置为secondary节点

# drbdadm secondary example

将另一台机器设置为primary节点 （在另一台机器上执行）

# drbdadm primary example

挂载文件系统，查看数据是否同步

# mount /dev/drbd0 /var/www/html

Note: 两台主机上的/dev/drbd0不能同时挂载， 只有状态为primary时，才能被挂载使用，而此时另一方的状态为secondary

**linux下磁盘镜像软件DRBD的使用**

**一、 什么是DRBD**

DRBD的全称为：Distributed Replicated Block Device (DRBD)分布式块设备复制,DRBD是由内核模块和相关脚本而构成，用以构建高可用性的集群。其实现方式是通过网络来镜像整个设备。它允许用户在远程机器上建立一个本地块设备的实时镜像。与心跳连接结合使用，也可以把它看作是一种网络RAID。

**二、DRBD是如何工作的** Drbd 负责接收数据，把数据写到本地磁盘，然后发送给另一个主机。另一个主机再将数据存到自己的磁盘中。目前，drbd 每次只允许对一个节点进行读写访问，这对于通常的故障切换高可用性集群来讲已经足够用了。以后的版本将支持两个节点进行读写存取。

**三、 drbd与现在的HA集群的关系**

一个drbd系统由两个以上节点构成，与HA集群类似，也有主用节点和备用节点之分，在带有主要设备的节点上，应用程序和操作系统可以运行和访问drbd设备（/dev/nbX）。  
 在主节点写入的数据通过drbd设备存储到主节点的磁盘设备中，同时，这个数据也会自动发送到备用节点相应的drbd设备，最终写入备用节点的磁盘设备中，在备用节点上，drbd只是将数据从drbd设备写入到备用节点的磁盘设备中。  
 大部分现行高可用性集群都会使用共享存储，而Drbd也可以作为一个共享存储设备，使用drbd不需要任何硬件的投资。因为它在IP网络中运行，所以，利用drbd作为共享存储设备，要节约很多成本，因为在价格上IP网络要比专用的存储网络经济的多。

**四、 DRBD实现原理图**DRBD是linux的内核的存储层中的一个分布式存储系统，可用使用DRBD在两台linux服务器之间共享块设备，共享文件系统和数据。类似于一个网络RAID1的功能，如图1所示：

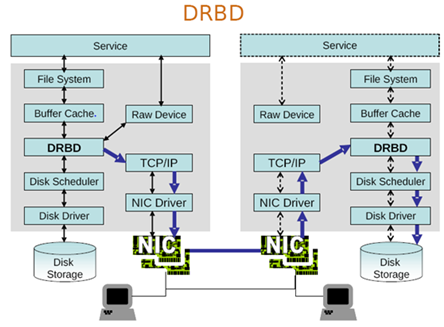
[](http://img1.51cto.com/attachment/201106/085505903.png)

图1

**五、DRDB的安装**从官方网站下载源码包来编译或直接使用yum源来安装，这里以CentOS为例说明安装过程，其它系统类似。  
[root@drbd1 ~]# uname -a  
Linux drbd1 2.6.18-194.11.1.el5 #1 SMP Tue Aug 10 19:09:06 EDT 2010 i686 i686 i386 GNU/Linux  
通过yum安装DRBD服务：  
[root@drbd1 ~]# yum -y install kmod-drbd83 drbd83  
检查DRBD是否安装成功：  
[root@drbd1 ~]# lsmod | grep -i drbd  
drbd                  228528  3   
[root@drbd1 ~]#  modprobe -l | grep -i drbd  
/lib/modules/2.6.18-194.11.1.el5/weak-updates/drbd83/drbd.ko  
安装成功之后/sbin目录下面有drbdadm，drbdmeta，drbdsetup命令，以及/etc/init.d/drbd启动脚本。

**六、配置DRDB**DRBD运行需要读取/etc/drbd.conf配置文件，下面是两台主机节点配置的drbd.conf文件的简单示例：  
[root@drbd1 ~]#cat /etc/drbd.conf  
#  
# drbd.conf  
#  
# create by [jackbillow@gmail.com](mailto:jackbillow@gmail.com) at 2010-08-12  
global {  
    # minor-count 64;  
    # dialog-refresh 5; # 5 seconds  
    # disable-ip-verification;  
usage-count no;     
#是否参加DRBD使用者统计，默认yes  
}

common {  
  syncer { rate 200M; }  
#设置主备节点同步时的网络速率最大值，单位是字节。  
}

resource r0 {  
  #资源名字为r0.  
  protocol      C;  
# 使用drbd的第三种同步协议,表示收到远程主机的写入确认后,则认为写入完成.  
  handlers {  
    pri-on-incon-degr "echo o > /proc/sysrq-trigger ; halt -f";  
    pri-lost-after-sb "echo o > /proc/sysrq-trigger ; halt -f";  
    local-io-error "echo o > /proc/sysrq-trigger ; halt -f";  
    fence-peer "/usr/lib64/heartbeat/drbd-peer-outdater -t 5";  
    pri-lost "echo pri-lost. Have a look at the log files. | mail -s 'DRBD Alert' root";  
    split-brain "/usr/lib/drbd/notify-split-brain.sh root";  
    out-of-sync "/usr/lib/drbd/notify-out-of-sync.sh root";  
  }

  net {  
    # timeout           60;  
    # connect-int       10;  
    # ping-int          10;  
    # max-buffers     2048;  
    # max-epoch-size  2048;  
    cram-hmac-alg "sha1";  
shared-secret "MySQL-HA";  
# DRBD同步时使用的验证方式和密码信息。  
  }

  disk {  
    on-io-error detach;  
fencing resource-only;  
#使用dpod功能（drbd outdate-peer daemon)保证在数据不同步时不进行切换。  
  }

  startup {  
    wfc-timeout 120;  
    degr-wfc-timeout 120;  
  }

  device        /dev/drbd0;  
   
  on dbm157 {  
#每个主机的说明以on开头,后面是hostname（uname -n），在后面的{}中为这个主机的配置。  
disk        /dev/sda2;  
#/dev/drbd0使用的磁盘分区是/dev/sda2。  
address     192.168.0.157:7788;  
#设置DRBD的监听端口,用于与另一台主机通信。  
    meta-disk   internal;  
  }  
  on dbm158 {  
disk        /dev/sda2;  
#/dev/drbd0使用的磁盘分区是/dev/sda2。  
address     192.168.0.158:7788;  
#设置DRBD的监听端口,用于与另一台主机通信。  
    meta-disk   internal;  #drbd的元数据存放方式。  
  }

}  
将上面这个drbd.conf文件分别复制到两台主机的/etc目录下。drbd.conf的配置参数很多，有兴趣的话可以使用命令：man drbd.conf来查看了解更多的参数说明。

**七、启动DRBD**

1 在两个节点执行  
在启动DRBD之前,你需要分别在两台主机的hdb1分区上,创建供DRBD记录信息的数据块.分别在两台主机上执行:   
[root@drbd1 ~]# drbdadm create-md r0 或者执行drbdadm create-md all  
[root@drbd2 ~]# drbdadm create-md r0

2在两个节点启动服务  
[root@drbd1 ~]#/etc/init.d/drbd start  
[root@drbd2 ~]#/etc/init.d/drbd start  
最好同时启动

3在任意节点查看节点状态  
[root@drbd1 ~]# cat /proc/drbd  
 1: cs:Connected ro:Secondary/Secondary ds:Inconsistent/Inconsistent C r----  
    ns:0 nr:0 dw:0 dr:0 al:0 bm:0 lo:0 pe:0 ua:0 ap:0 ep:1 wo:b oos:2007644  
对输出的含义解释如下：  
ro表示角色信息，第一次启动drbd时，两个drbd节点默认都处于Secondary状态，  
ds是磁盘状态信息，“Inconsistent/Inconsisten”，即为“不一致/不一致”状态，表示两个节点的磁盘数据处于不一致状态。  
Ns表示网络发送的数据包信息。  
Dw是磁盘写信息  
Dr是磁盘读信息

4设置主节点  
由于默认没有主次节点之分，因而需要设置两个主机的主次节点，选择需要设置为主节点的主机，然后执行如下命令：  
[root@drbd1 ~]#drbdsetup /dev/drbd0 primary –o  
或者执行下面命令也是可以的  
[root@drbd1 ~]#drbdadm -- --overwrite-data-of-peer primary all  
第一次执行完此命令后，在后面如果需要设置哪个是主节点时，就可以使用另外一个命令：  
[root@drbd1 ~]#/sbin/drbdadm primary r0或者/sbin/drbdadm primary all  
执行此命令后，开始同步两台机器对应磁盘的数据  
[root@drbd1 ~]#cat /proc/drbd  
1: cs:SyncSource ro:Primary/Secondary ds:UpToDate/Inconsistent C r----  
    ns:576224 nr:0 dw:0 dr:581760 al:0 bm:34 lo:84 pe:369 ua:256 ap:0 ep:1 wo:b oos:1443196  
        [====>...............] sync'ed: 28.4% (1443196/2007644)K delay\_probe: 69  
        finish: 0:03:56 speed: 6,024 (5,876) K/sec  
从输出可知：  
 “ro状态现在变为“Primary/Secondary”，“ds”状态也变为“UpToDate/Inconsistent”，也就是“实时/不一致”状态，现在数据正在主备两个主机的磁盘间进行同步，且同步进度为28.4%，同步速度每秒5.8M左右。  
等待片刻，再次查看同步状态，输出如下：  
[root@drbd1 ~]#cat /proc/drbd  
1: cs:Connected ro:Primary/Secondary ds:UpToDate/UpToDate C r----  
ns:2007644 nr:0 dw:0 dr:2007644 al:0 bm:123 lo:0 pe:0 ua:0 ap:0 ep:1 wo:b oos:0  
可以看到同步完成了，并且“ds“状态也变为“UpToDate/UpToDate”了。即为“实时/实时”状态了。  
如果第一次设置主备节点时使用“/sbin/drbdadm primary r0”命令，那么会提示如下错误：  
0: State change failed: (-2) Need access to UpToDate data  
Command '/sbin/drbdsetup 0 primary' terminated with exit code 17  
只要第一次用上面命令成功后，以后就可以用“/sbin/drbdadm primary r0”命令了。  
5格式化文件系统  
由于mount操作只能在主节点进行，所以只有设置了主节点后才能格式化磁盘分区，然后挂载：  
[root@drbd1 ~]#mkfs.ext3 /dev/drbd0   
[root@drbd1 ~]#mount /dev/drbd0 /mnt

**八、 DRBD主备节点切换**主备节点切换有两种方式，分别是停止drbd服务切换和正常切换，依次介绍：

1停止drbd服务切换  
关闭主节点服务，此时挂载的drbd分区就自动在主节点卸载了，然后在备用节点执行切换命令：  
[root@drbd2 ~]#drbdadm primary all  
此时会报错：  
2: State change failed: (-7) Refusing to be Primary while peer is not outdated  
Command 'drbdsetup 2 primary' terminated with exit code 11  
因此，必须在备用节点执行如下命令：  
[root@drbd2 ~]#drbdsetup /dev/drbd0 primary –o  
或者  
[root@drbd2~]#drbdadm -- --overwrite-data-of-peer primary all  
此时就可以正常切换了。  
当在备用节点执行切换到主节点命令后，原来的主用节点自动变为备用节点。无需在主用节点再次执行切换到备用节点的命令。

2正常切换  
在主节点卸载磁盘分区，然后执行  
[root@drbd1 ~]#drbdadm secondary all  
如果不执行这个命令，直接在备用节点执行切换到主节点的命令，会报错：  
2: State change failed: (-1) Multiple primaries not allowed by config  
Command 'drbdsetup 2 primary' terminated with exit code 11  
接着，在备用节点执行  
[root@drbd2 ~]#drbdadm primary all  
最后在备用节点挂载磁盘分区即可：  
[root@drbd2 ~]#mount /dev/drbd2  /mnt

**DRBD+Heartbeat**

**DRBD简介**  
　　Distributed Replicated Block Device(DRBD)是一个用软件实现的、无共享的、服务器之间镜像块设备内容的存储复制解决方案。

     数据镜像：实时、透明、同步（所有服务器都成功后返回）、异步（本地服务器成功后返回）

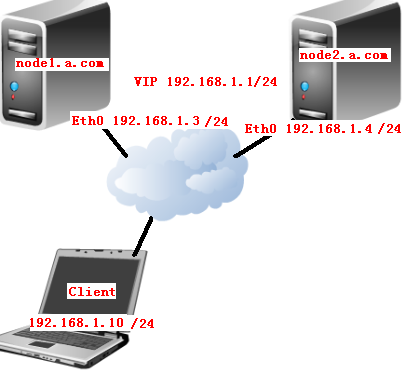
     DBRD的核心功能通过Linux的内核实现，最接近系统的IO栈，但它不能神奇地添加上层的功能比如检测到EXT3文件系统的崩溃。

    DBRD的位置处于文件系统以下，比文件系统更加靠近操作系统内核及IO栈。

   以上信息，来自于互联网，请尊重版权。接下来笔者将向您展示如何配置heartbeat+drbd的实例。

**环境要求：某企业为了实现高可用性的群集文件系统，出于成本的考虑采用drbd技术。**

**实验拓扑：**

[](http://img1.51cto.com/attachment/201205/010844506.png)

**实验步骤：**

**一、准备环境**

**1.系统环境**

# **uname –r** //内核版本   
**2.6.18-164.el5**   
# **cat /etc/redhat-release**  //查看系统信息   
Red Hat Enterprise Linux Server release **5.4** (Tikanga)

**2.修改主机名称等**

在节点1上进行修改

# **hostname node1.a.com** //修改主机名称   
# **vim /etc/sysconfig/network** //修改文件

NETWORKING=yes   
NETWORKING\_IPV6=no   
HOSTNAME=**node1.a.com**

之后退出再次登陆。

# **hostname** //查看名称   
**node1.a.com**   
# **vim /etc/hosts** //修改hosts文件

127.0.0.1                localhost.localdomain localhost   
::1             localhost6.localdomain6 localhost6   
**192.168.1.3 node1.a.com   
192.168.1.4 node2.a.com**

在节点2上进行修改

# **hostname node2.a.com** //修改主机名称   
# **vim /etc/sysconfig/network** //修改文件

NETWORKING=yes   
NETWORKING\_IPV6=no   
**HOSTNAME=node2.a.com**

之后退出再次登陆。

# **hostname** //查看名称   
node2.a.com   
# **vim /etc/hosts** //修改hosts文件

127.0.0.1                localhost.localdomain localhost   
::1             localhost6.localdomain6 localhost6   
**192.168.1.3 node1.a.com   
192.168.1.4 node2.a.com**

**3.安装所需的文件**

预先将所需的文件上传到管理员家目录。以下操作在node1.a.com和node2.a.com上都进行操作。

为了方便你的使用，笔者已将全部所需文件上传到笔者空间，提供您免费下载：<http://down.51cto.com/data/401397>

# **ll**   
total 3084   
-rw------- 1 root root    1291 Feb  8 02:05 anaconda-ks.cfg   
**-rw-r--r-- 1 root root  221868 May  7  2012 drbd83-8.3.8-1.el5.centos.i386.rpm   
-rw-r--r-- 1 root root 1637238 Mar 14  2010 heartbeat-2.1.4-9.el5.i386.rpm   
-rw-r--r-- 1 root root  293349 Mar 14  2010 heartbeat-devel-2.1.4-9.el5.i386.rpm   
-rw-r--r-- 1 root root  230890 Mar 14  2010 heartbeat-gui-2.1.4-9.el5.i386.rpm   
-rw-r--r-- 1 root root  111742 Mar 14  2010 heartbeat-ldirectord-2.1.4-9.el5.i386.rpm   
-rw-r--r-- 1 root root   92070 Mar 14  2010 heartbeat-pils-2.1.4-10.el5.i386.rpm   
-rw-r--r-- 1 root root  179199 Mar 14  2010 heartbeat-stonith-2.1.4-10.el5.i386.rpm**   
-rw-r--r-- 1 root root   35236 Feb  8 02:04 install.log   
-rw-r--r-- 1 root root    3995 Feb  8 02:02 install.log.syslog   
**-rw-r--r-- 1 root root  125974 May  7  2012 kmod-drbd83-8.3.8-1.el5.centos.i686.rpm   
-rw-r--r-- 1 root root   56817 Mar 14  2010 libnet-1.1.4-3.el5.i386.rpm   
-rw-r--r-- 1 root root   92071 Mar 14  2010 perl-MailTools-1.77-1.el5.noarch.rpm**   
# **yum localinstall \*.rpm --nogpgcheck –y** //使用yum进行安装可以有效的解决包的依赖性问题。

**4.同步时钟**

要求两个节点上的始终必须要一致。

# **hwclock -s** //同步始终   
# **date** //查看日期   
Wed Feb  8 03:38:44 CST 2012   
#

**5.加载模块**

在两个节点上都去进行如下操作。

# **modprobe drbd** //加载drbd模块   
#

# **lsmod |grep drbd** //查看模块   
drbd                  228528  0   
#

**6.创建新分区**

以下操作要求在node1.a.com和node2.a.com上都进行。同时要求两个节点所创建的新分区大小一致。

# **fdisk -l** //查看有关情况

Disk /dev/sda: 21.4 GB, 21474836480 bytes   
255 heads, 63 sectors/track, 2610 cylinders   
Units = cylinders of 16065 \* 512 = 8225280 bytes

   Device Boot      Start         End      Blocks   Id  System   
/dev/sda1   \*           1          13      104391   83  Linux   
/dev/sda2              14        1288    10241437+  83  Linux   
/dev/sda3            1289        1415     1020127+  82  Linux swap / Solaris   
# **fdisk /dev/sda**

The number of cylinders for this disk is set to 2610.   
There is nothing wrong with that, but this is larger than 1024,   
and could in certain setups cause problems with:   
1) software that runs at boot time (e.g., old versions of LILO)   
2) booting and partitioning software from other OSs   
   (e.g., DOS FDISK, OS/2 FDISK)

Command (m for help): **n** //创建新分区   
Command action   
   e   extended   
   p   primary partition (1-4)   
**p** //创建主分区   
Selected partition 4   
First cylinder (1416-2610, default 1416):   
Using default value 1416   
Last cylinder or +size or +sizeM or +sizeK (1416-2610, default 2610): **+1G** //两个节点的分区大小要一致。

Command (m for help): **p** //显示

Disk /dev/sda: 21.4 GB, 21474836480 bytes   
255 heads, 63 sectors/track, 2610 cylinders   
Units = cylinders of 16065 \* 512 = 8225280 bytes

   Device Boot      Start         End      Blocks   Id  System   
/dev/sda1   \*           1          13      104391   83  Linux   
/dev/sda2              14        1288    10241437+  83  Linux   
/dev/sda3            1289        1415     1020127+  82  Linux swap / Solaris   
/dev/sda4            1416        1538      987997+  83  Linux

Command (m for help): **w** 保存

# **fdisk -l** //再次查看

Disk /dev/sda: 21.4 GB, 21474836480 bytes   
255 heads, 63 sectors/track, 2610 cylinders   
Units = cylinders of 16065 \* 512 = 8225280 bytes

   Device Boot      Start         End      Blocks   Id  System   
/dev/sda1   \*           1          13      104391   83  Linux   
/dev/sda2              14        1288    10241437+  83  Linux   
/dev/sda3            1289        1415     1020127+  82  Linux swap / Solaris   
**/dev/sda4            1416        1538      987997+  83  Linux**   
#

# **partprobe /dev/sda** //重新加载

**二、DRBD配置**

以下操作要求在节点node1和node2上都做。

**1.修改conf配置文件**

# **cd /usr/share/doc/drbd83-8.3.8/**#**ls**   
ChangeLog  COPYING **drbd.conf**  file.list  README

#**cp drbd.conf /etc/** 复制配置文件到相关目录   
cp: overwrite `/etc/drbd.conf'? y   
#**cd /etc/drbd.d/**   
# **ls**   
global\_common.conf   
# **cp -p global\_common.conf global\_common.conf.bak** //最好对文件进行备份   
# **vim global\_common.conf**

//第一列是行的序号

1 global {   
2         usage-count no;   
3 }   
4   
5 common {   
6         protocol C;   
7         startup {

8                 wfc-timeout 120;

9                 degr-wfc-timeout 120;

10         }

11         disk {

12                 on-io-error detach;

13                 fencing resource-only;

14         }

15         net {

16         cram-hmac-alg "sha1";

17         shared-secret "mydrbdlab";         
18         }   
19         syncer {   
20                 rate 100M;   
21         }   
22 }

**2.修改资源文件**

# **cd /etc/drbd.d/**

# **vim web.res**

1 resource web {   
2        on node1.a.com {   
3        device /dev/drbd0;   
4        disk /dev/sda4;   
5        address 192.168.1.3:7789;   
6        meta-disk  internal;   
7        }   
8        on node2.a.com {   
9        device /dev/drbd0;   
10        disk /dev/sda4;   
11        address 192.168.1.4:7789;   
12        meta-disk  internal;   
13        }   
14 }

**3.创建资源**

#**drbdadm create-md web**   
Writing meta data...   
initializing activity log   
NOT initialized bitmap   
New drbd meta data block successfully created.

**4.启动服务、测试**

# **service drbd start**   
Starting DRBD resources: drbdsetup 0 show:5: delay-probe-volume **0k =&gt; 0k out of range** [4..1048576]k.   
# **drbdadm adjust web** //测试   
**drbdsetup 0 show:5: delay-probe-volume 0k =&gt; 0k out of range [4..1048576]k**.   
查看服务的情况

# **service drbd status**   
drbd driver loaded OK; device status:   
version: 8.3.8 (api:88/proto:86-94)   
GIT-hash: d78846e52224fd00562f7c225bcc25b2d422321d build by mockbuild@builder10.centos.org, 2010-06-04 08:04:16   
m:res  cs         ro                   ds                         p  mounted  fstype   
0:web  Connected  **Secondary/Secondary**  Inconsistent/Inconsistent  C   
当然也可以使用下面的指令进行查看。

# **drbd-overview**   
  0:web  Connected **Secondary/Secondary** Inconsistent/Inconsistent C r----

**5.指定主设备**

以下操作要求只是在一个节点上进行完成。笔者只在node1.a.com上进行完成，指定node1为主要的节点

# **drbdadm -- --overwrite-data-of-peer primary web** //指定node1为资源主节点

# **watch -n 1 'cat /proc/drbd'** //使用此命令，可以动态的查看具体的同步过程

# **mkfs -t ext3 -L drbdweb /dev/drbd0** //格式化

# **mkdir /web**   
# **mount /dev/drbd0 /web** //挂载   
#

# **service drbd status //在node1.a.com**上查看状态   
drbd driver loaded OK; device status:   
version: 8.3.8 (api:88/proto:86-94)   
GIT-hash: d78846e52224fd00562f7c225bcc25b2d422321d build by mockbuild@builder10.centos.org, 2010-06-04 08:04:16   
m:res  cs         ro                 ds                 p  mounted  fstype   
0:web  Connected  **Primary/Secondary**  UpToDate/UpToDate  C  /web     ext3   
#

切换到node2.a.com上进行查看

# **service drbd status**   
drbd driver loaded OK; device status:   
version: 8.3.8 (api:88/proto:86-94)   
GIT-hash: d78846e52224fd00562f7c225bcc25b2d422321d build by mockbuild@builder10.centos.org, 2010-06-04 08:04:16   
m:res  cs         ro                 ds                 p  mounted  fstype   
0:web  Connected  **Secondary/Primary**  UpToDate/UpToDate  C   
#

**三、NFS配置**

两台服务器的nfs配置必须要一致。

**1.修改相关文件**

# **vim /etc/exports**

**/web \*(rw,sync,insecure,no\_root\_squash,no\_wdelay)**

**2.执行相关操作。**

#**service portmap start && chkconfig portmap on**

#**service nfs start && chkconfig nfs on**

**3.修改nfs启动脚本**

# **vim /etc/init.d/nfs**

122         killproc nfsd –9

**四、Heartbeat配置**

Heartbeat的配置要求两个节点的配置一致。在node1和node2上进行如下操作

**1.复制模版文件**

# **cd /usr/share/doc/heartbeat-2.1.4/**# **cp authkeys ha.cf haresources /etc/ha.d/**

**2.修改相关的配置信息**

# **cd /etc/ha.d/** //切换到相关目录   
# **vim ha.cf**

24 debugfile /var/log/ha-debug

29 logfile /var/log/ha-log

34 logfacility     local0

48 keepalive 2

56 deadtime 10

76 udpport 694

121 bcast eth0 //此行可以添加在任意一行

157 auto\_failback off

211 node    node1.a.com   
212 node    node2.a.com

**3.修改资源文件**

# **echo "node1.a.com IPaddr::192.168.1.1/24/eth0 drbddisk::web Filesystem::/dev/drbd0::/web::ext3 killnfsd"&gt;&gt;/etc/ha.d/haresources**

**4.修改key文件**

# **vim authkeys**

23 auth 1   
24 1 crc

**5.手工创建文件**

# **cd /etc/ha.d/resource.d/**# **echo "killall -9 nfsd ; /etc/init.d/nfs restart ; exit 0" &gt;&gt;/etc/ha.d/resource.d/killnfsd**

**6.修改配置文件的权限**

# **chmod 600 /etc/ha.d/authkeys**   
# **chmod 755 /etc/ha.d/resource.d/killnfsd**

**7.启动服务**

# **service heartbeat start**

**五、测试**

使用客户端进行测试

**1.挂载**

#**mkdir /mnt/nfs**   
#**mount 192.168.1.1:/web /mnt/nfs**   
#**mount 192.168.1.1:/web /mnt/nfs**

**2.创建测试脚本文件**

#**vim /mnt/test.sh**//此配置脚本文件的主要作用是为了反复的读写操作

while true   
do   
echo --\&gt;trying touch x:`date`  
touch x   
echo \<-----done touch x:`date`  
echo   
sleep 2   
done

**3.执行测试文件**

#**cd /mnt/nfs/**   
# **bash /mnt/test.sh**

--->trying touch x:Sun May 6 07:49:20 CST 2012  
<-----done touch x:Sun May 6 07:49:20 CST 2012

--->trying touch x:Sun May 6 07:49:22 CST 2012  
<-----done touch x:Sun May 6 07:49:22 CST 2012

--->trying touch x:Sun May 6 07:49:24 CST 2012  
<-----done touch x:Sun May 6 07:49:24 CST 2012

--->trying touch x:Sun May 6 07:49:26 CST 2012  
<-----done touch x:Sun May 6 07:49:26 CST 2012

--->trying touch x:Sun May 6 07:49:28 CST 2012  
<-----done touch x:Sun May 6 07:49:30 CST 2012

--->trying touch x:Sun May 6 07:49:32 CST 2012  
<-----done touch x:Sun May 6 07:49:32 CST 2012

--->trying touch x:Sun May 6 07:49:34 CST 2012  
<-----done touch x:Sun May 6 07:49:34 CST 2012

--->trying touch x:Sun May 6 07:49:36 CST 2012  
<-----done touch x:Sun May 6 07:49:36 CST 2012

//让客户端一直执行脚本文件，到服务器节点node1上关闭heartbeat服务，之后在客户端会发现丢弃现象，之后文件系统又恢复正常。

--->trying touch x:Sun May 6 07:50:22 CST 2012  
<-----done touch x:Sun May 6 07:50:22 CST 2012

--->trying touch x:Sun May 6 07:50:24 CST 2012  
touch: cannot touch `x': Stale NFS file handle  
<-----done touch x:Sun May 6 07:50:30 CST 2012

--->trying touch x:Sun May 6 07:50:33 CST 2012  
touch: cannot touch `x': Stale NFS file handle  
<-----done touch x:Sun May 6 07:50:33 CST 2012

--->trying touch x:Sun May 6 07:50:35 CST 2012  
touch: cannot touch `x': Stale NFS file handle  
<-----done touch x:Sun May 6 07:50:35 CST 2012

--->trying touch x:Sun May 6 07:50:37 CST 2012  
<-----done touch x:Sun May 6 07:50:37 CST 2012

--->trying touch x:Sun May 6 07:50:39 CST 2012  
<-----done touch x:Sun May 6 07:50:39 CST 2012