使用 DRBD 备份 Mysql

日期	说明	联系邮箱
2011-1-13	描述了跨 IDC 尝试用 DRBD 对 Mysql 进行备份的案例	sysops@mail2000.us

目 录

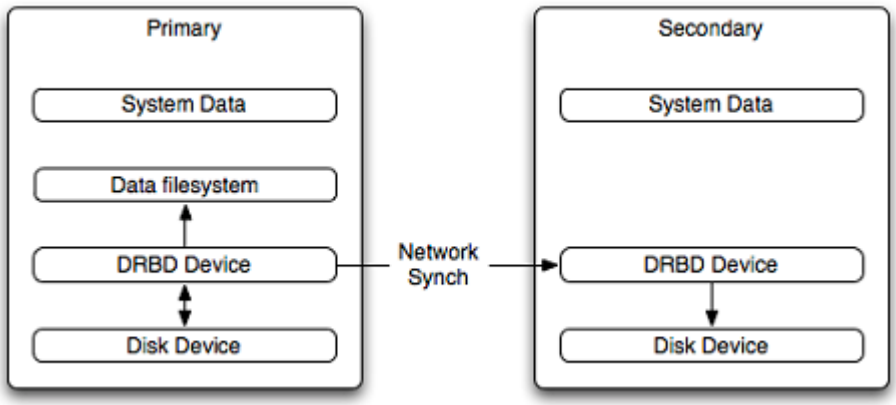
1 安装概述

在 Ubuntu 9.10 上安装 DRBD 很简单，执行一个命令即可：

```
apt-get install drbd8-utils
```

它会自动安装 drbd 内核模块，和一些管理工具，如/sbin 下的 drbdadm、drbdmeta、drbdsetup，和/etc/init.d/drbd 启动脚本。

DRBD 的工作原理在于创建/dev/drdbX 的逻辑块设备，并将逻辑块设备映射到底层的物理块设备（如硬盘分区）。DRDB 的网络 IO 层操作逻辑块设备，并保持数据在主从之间同步。物理 IO 层将逻辑块设备的内容写入到磁盘的物理块设备上。



2 配置使用

2.1 网络环境

测试的网络环境如下：

主机名	IP 地址	位置	角色
drbd01	202.104.186.180	珠海	主服务
drbd02	121.9.221.222	茂名	从服务

要在/etc/hostname 和/etc/hosts 两个文件里都定义好主机名和 IP 地址。例如 drbd01 的：

root@drbd01:~# cat /etc/hostname
drbd01
root@drbd01:~# cat /etc/hosts
202.104.186.180 drbd01

121.9.221.222　　drbd02

2.2 硬件环境

在配置 DRBD 之前，要在主机上腾出一个块设备，块设备就是系统里一个磁盘分区。这个分区通常都已有文件系统了（例如 ext4）。但最好是不要有文件系统，所以在新装机器做规划时，留一个分区不要格式化成文件系统。

在测试环境里，drdb01 的/dev/sda4，drbd02 的/dev/sda7 都是 mount 在/data 分区，之前没有数据，可以使用。先把这两个/data 分区卸载掉，执行：

umount /data

由于这两个分区之前都是有文件系统的，所以 umount 掉后，还要用 dd 清空块设备开头一些数据：

dd if=/dev/zero bs=512 count=512 of=/dev/sda4
dd if=/dev/zero bs=512 count=512 of=/dev/sda7

根据测试机的实际情况，第一句在 drdb01 上执行，第二句在 drdb02 上执行。不这样做的话，后面 drbdadm create-md 命令执行不了。所以物理块设备最好是未格式化过的分区。

2.3 配置 DRBD

DRDB 只有一个配置文件：/etc/drbd.conf，测试机的内容如下：

```
global { usage-count no; }  # 使用情况不通知厂商
common { syncer { rate 100M; } }  # 同步速率 100M
resource r0 {  # 创建 r0 这个资源
        protocol C;  # C 代表完全同步，数据在两边都写到物理磁盘才返回
        startup {  # 启动时的一些时间参数
                wfc-timeout  15;
                degr-wfc-timeout 60;
        }
        net {  # 网络通信参数
                cram-hmac-alg sha1;  # 主从身份验证的哈希算法
                shared-secret "pay.duowan";  # 验证共享字符串
        }
        on drbd01 {  # 创建 drbd01 节点
                device /dev/drbd0;  # 逻辑块设备
                disk /dev/sda4;  # 物理块设备
                address 202.104.186.180:7788;  # drdb01 的 IP 地址和端口
                meta-disk internal;  # meta-disk 为 internal
```

```
        }
        on drbd02 {   # 创建 drbd02 节点，其他内容与 drbd01 同
                device /dev/drbd0;
                disk /dev/sda7;
                address 121.9.221.222:7788;
                meta-disk internal;
        }
}
```

这个配置文件在主从机上都必须完全一致。

创建完配置文件后，就可以启动 drbd，在主从上都运行：

```
/etc/init.d/drbd start
```

然后 lsmod |grep drbd 看看内核模块是否加载。

接下来，用 drbdadm 工具初始化元数据，在主从上都运行：

```
drbdadm create-md r0
```

初始化成功后，在 drbd01 上运行如下命令（语法有点奇怪），表示把 drbd01 设置为 Primary，并且同步数据到从机：

```
drbdadm -- --overwrite-data-of-peer primary all
```

至此，DRBD 就配置完成了。

2.4 使用 DRBD

在主机 drbd01 上，可以 mount 逻辑设备成为文件系统，执行命令如下：

```
mkfs.ext3 /dev/drbd0
mount /dev/drbd0 /drbd
```

表示把/dev/drbd0 格式化成 ext3 文件系统（当然 ext4 也可以），然后把该设备 mount 到/drbd 目录下。这个就是我们要使用的数据目录，mysql 的 datadir 应该放在该目录下。

往/drbd 目录里写入的任何文件，都同步到从机上。这个同步是执行的块设备拷贝，而不是文件拷贝。但是，在从机上不能 mount 逻辑块设备，也就是说，从机上到底有些什么数据，平时是看不到的。要使用从机的数据，必须做如下切换：

（1）在主机 drbd01 上，卸载掉/drbd 目录：

```
umount /drbd
```

把主服务变成从服务：

```
drbdadm secondary r0
```

（2）在从机 drbd02 上，把从服务变成主服务：

```
drbdadm primary r0
```

挂载逻辑设备：

```
mount /dev/drbd0 /drbd
```

然后，在从机上就可以访问到/drbd 目录里面的文件，内容跟主机的一致。

2.5 在 DRBD 上运行 Mysql

2.5.1 Mysql 主机

Mysql 主机运行在 drbd01 上。在 DRBD 上运行 Mysql 没有特殊技巧，在配置好 DRBD 环境后，格式化 drbd 文件系统，并把它挂载到系统的一个目录里。然后，把 mysql 的数据目录配置为已挂载的 drbd 目录。

如前所述，已格式化/dev/drbd0 为 ext3 文件系统，并且挂载在/drbd 目录里。那么，任何在/drbd 目录的文件变更操作，都会同步到远程文件系统。在 apt-get 安装完 mysql 后，修改 /etc/mysql/my.cnf，指定 datadir 为/drbd：

```
datadir = /drbd/mysql
```

当然，要保证/drbd/mysql 目录存在，并设置了正确的属主：

chown –R mysql:mysql /drbd/mysql

然后，初始化 Mysql 数据库、创建系统表，运行如下命令：

mysql_install_db --user=mysql

因为用的是 Ubuntu 系统，出于安全考虑，ubuntu 对系统服务需要访问的目录设置了读写限制。所以在运行上述命令之前，要修改/etc/apparmor.d/usr.sbin.mysqld 文件，设置目录许可，并重启 apparmor 服务。

最后，启动 mysql 数据库，执行如下命令：

/etc/init.d/mysql start

在启动的过程中，会自动创建 Innodb 表空间（如果配置了 Innodb 的话）。我测试的 Innodb 表空间配置如下：

```
innodb_data_file_path = ibdata1:10G;ibdata2:10G;ibdata3:10G:autoextend
```

实际情况是，创建每个 10G 的表空间文件，要用一个小时：

```
-rw-rw---- 1 mysql mysql 10737418240 2011-01-17 12:32 ibdata1
-rw-rw---- 1 mysql mysql 10737418240 2011-01-17 13:30 ibdata2
```

如上，第一个文件创建完的时间是 12:32，第二个文件创建完的时间是 13:30，基本一个小时。这也说明跨 IDC 使用 DRBD 是如此之慢。

2.5.2 Mysql 从机

Mysql 从机运行在 drbd02 上。Mysql 从机不需要特别配置，例如创建表空间等过程都不需要，因为文件是自动从主机上同步过来的。要保证 Mysql 主从的 my.cnf 配置文件完全一致。另外，两台机的系统环境需要严格一致，例如 mysql 的用户 ID 和组 ID 都必须数字相等，否则就会有问题。

2.5.3 数据库同步

Mysql 完全起来后，随便创建一个库，并创建一个测试表：

```
CREATE TABLE `myt` (
  `id` int(11) NOT NULL AUTO_INCREMENT,
  `name` varchar(16) DEFAULT NULL,
  `age` tinyint(3) DEFAULT NULL,
  `sex` enum('F','M') DEFAULT NULL,
  `addr` varchar(64) DEFAULT NULL,
  `postcode` int(5) DEFAULT NULL,
  PRIMARY KEY (`id`)
) ENGINE=InnoDB DEFAULT CHARSET=utf8;
```

往这个表里插入 100 万行数据，共耗时 6 分 14 秒。数据内容类似如下：

```
+--------+----------+------+------+----------------------------------+----------+
| id     | name     | age  | sex  | addr                             | postcode |
+--------+----------+------+------+----------------------------------+----------+
| 380485 | e_MmBSLf |   68 | F    | 0yzqFpfmQy3SmoW_3nmT-BYTa2HkqR3U |    33511 |
| 380486 | TupJiy4E |   51 | M    | gxbiLvuDUQn_JJaENFU9CmdDblUiX5hD |    90224 |
| 380487 | Fb2aGE0G |   63 | F    | v7nfkoR8sqAoNCJXqenhP_zM6cx5TfEy |    41116 |
+--------+----------+------+------+----------------------------------+----------+
```

写完数据后，在 drbd02 上，执行如下命令：

drbdadm disconnect r0
drbdadm primary r0

第一个命令表示断开 DRBD，第二个命令表示把从机的 DRBD 提升为主机状态。如果不执行第一个命令的话，则第二个命令执行不了。这里仅仅是为了测试，正常来说，应该是把主设为从，再把从设为主，不用断开服务。

然后 mount 从机的文件系统：

mount /dev/drbd0 /drbd

并启动 mysql：

/etc/init.d/mysql start

登录进数据库，执行 checksum table，确认主机和从机的数据库内容完全一致。

2.6 需求实例

2.6.1 需求

假设 drbd01 为主数据库，位于机房 A；drbd02 为从数据库，位于机房 B。某天机房 A 完全挂掉了，不可访问，需要 B 机房的 drbd02 顶替服务。在 A 机房恢复后，再将服务切换回去。

2.6.2 操作步骤

在 A 机房挂掉后，此时唯一能操作的是 drbd02，在 drbd02 上执行操作步骤如下：

（1） 执行/etc/init.d/drbd status，此时看到 drbd 已处于断开状态，因此可切换到主。
（2） 运行 drbdadm primary all，将 drbd02 提升为主服务。
（3） mount /dev/drbd0 /drbd，将 DRBD 文件系统挂载上。
（4） 执行/etc/init.d/mysql start，启动 Mysql。

此时 drbd02 的 Mysql 就可以继续提供服务。

假设过了一段时间，机房 A 又恢复了，此时需要做的工作如下：

（1） 在 drbd01 上，执行 mysqladm shutdown 先关掉 mysql，然后 umount /drbd 卸载掉 DRBD 文件系统。
（2） 在 drbd01 上，执行 drbdadm secondary all; drbdadm -- --discard-my-data connect all 这 2 个命令，把 drbd01 设置为从服务，并且开始同步数据，并丢弃所有与 drbd02 不一致的数据。
（3） 在 drbd02 上，执行 drbdadm primary all; drbdadm connect all 这 2 个命令，把 drbd02 仍然设置为主服务，并且开始同步数据。
（4） 执行完第（2）和第（3）步后，两台数据库的内容就完全一致。上述两个步骤不能反过来做。

2.6.3 结论

在 drbd01 的数据内容与 drbd02 的完全同步一致后，如何将 drbd01 恢复为主服务呢？很遗憾，没有办法，除非把 drbd02 停机维护（或者用 iptables 拦截掉所有网络连接）。所以，DRBD 其实不适合我们的需求。

还有一个原因影响 DRBD 远程备份，就是性能问题。使用 DRBD 作为远程备份，数据库 IO 效率大大降低。只有一个请求在本地和远程都写完后，操作系统的写请求才算完成，这无疑非常慢。它的效率取决于网络 IO 的效率。而两个 IDC 之间网络速度是非常慢的（相对于磁盘速度），所以数据库写效率很低下。

DRBD 有三种同步协议：A、B 和 C。A 是在本地磁盘写完就算完成。B 是在本地磁盘写完，并且远程 Socket 接受完毕，就算完成。C 是本地和远程磁盘都写完，才算完成。适合我们的是 C，或者 B 也能接受，但无论 B 和 C 都非常慢，受网络 IO 影响。所以，DRBD 其实只适合于本地的同步备份，而不是远程。

3 管理操作

3.1 查看状态

运行命令：

/etc/init.d/drbd status　# 或者
cat /proc/drbd

都可以。关于输出的详细解释，请见：
http://www.drbd.org/docs/working/

3.2 drbdadm 管理命令

drbdadm 是用来执行 drbd 管理工作的，常用命令包括：

把本地设备设为主设备：

drbdadm primary all

把本地设备设为从设备：

drbdadm secondary all

临时断开本地 DRBD 节点：

drbdadm disconnect all

重新连接 DRBD 节点：

drbdadm connect all

其他更详细的用法，请见：
http://www.drbd.org/users-guide/re-drbdadm.html

3.3 关于 DRBD + Heartbeat

Mysql 官方很推荐使用 DRBD + Heartbeat 配置数据库高可用集群，但 Mysql 的文档写明了在本地做 DRBD 同步备份，而不是远程。引用如下：

Because of the nature of the DRBD system, the critical requirements are for a very fast exchange of the information between the two hosts. To ensure that your DRBD setup is available to switch over in the event of a failure as quickly as possible, you must transfer the information between the two hosts using the fastest method available.

而且在 Mysql 的 Heartbeat 配置里，对于 DRBD 也是没有开启 auto failback：

```
auto_failback    off
```

也就是说，在故障恢复后，主数据库不能自动恢复为主服务。

3.4 关于主从切换备忘

（1） 如果主机 mount 了 DRBD 目录，则不能切换为从，必须先 umount DRBD 文件系统。
（2） 如果主机没有切换为从，在通信不断开前提下，从机不能切换为主。
（3） 在通信断开（包括网络中断或者执行 drbdadm disconnect）前提下，从机可以无条件

切换为主。

（4） 在非主状态下，不能 mount DRBD 文件系统，所以从机必须先切换为主，数据库才能起来。

（5） 在通信断开又恢复后，很可能发现主从处在不同步的状态，不管怎么 drbdadm connect 都没用。如果处于这个状态，那么按如下操作做：

1. Determine which server has the most accurate data (call it A - call the other server B)
2. on B: drbdadm secondary all; drbdadm -- --discard-my-data connect all
3. on A: drbdadm primary all; drbdadm connect all