

## LSA-3 MySQL 主辅--一主多辅

mysql 最新的版本都需要cmake编译安装，估计以后的版本也会采用这种方式，所以特地记录一下安装步骤及过程，以供参考。

注意：此安装是默认CentOS下已经安装了最新工具包，比如GNU make, GCC, Perl, libncurses5-dev (redhat下叫ncurses)，如果在编译安装过程中发现有缺失的工具包，先yum install 单独下载安装再继续即可。

以下安装中涉及的几点需要提前说明的问题：

1. 所有下载的文件将保存在 /usr/local/src/ 目录下
2. mysql 将以mysql用户运行，而且将加入 service 开机自动运行
3. mysql 将被安装在 /usr/local/mysql/ 目录下
4. mysql 默认安装使用 utf8 字符集
5. mysql 的数据和日志文件保存在 /data/mysql/ 对应目录下
6. mysql 的配置文件保存于/etc/my.cnf

### 二. 安装 cmake

```
cd /usr/local/src
```

```
tar zxvf cmake-2.8.7.tar.gz
cd cmake-2.8.7
./bootstrap
gmake
gmake install
```

gmake是GNU Make的缩写。Linux系统环境下的make就是GNU Make，之所以有gmake，是因为在别的平台上，make一般被占用，GNU make只好叫gmake了。比如在安装二进制文件进行编译时要使用make命令，但如果在Solaris或其他非GNU系统中运行，必须使用GNU make，而不是使用系统自带的make版本，这时要用gmake代替make进行编译。

```
cd ../
```

```
# tar zxvf bison-2.5.tar.gz
# cd bison-2.5
# ./configure
# make
# make install
```

```
cd ../
```

### 三. 编译安装 MySQL 5.5.22

```
# /usr/sbin/groupadd mysql
# /usr/sbin/useradd -g mysql mysql

# tar xzvf mysql-5.5.22.tar.gz
# cd mysql-5.5.22/

# cmake -DCMAKE_INSTALL_PREFIX=/usr/local/mysql -
```

```
DMYSQL_UNIX_ADDR=/var/lib/mysql/mysql.sock -DDEFAULT_CHARSET=utf8 -  
DDEFAULT_COLLATION=utf8_general_ci -DWITH_EXTRA_CHARSETS=all -  
DWITH_MYISAM_STORAGE_ENGINE=1 -DWITH_INNOBASE_STORAGE_ENGINE=1 -  
DWITH_MEMORY_STORAGE_ENGINE=1 -DWITH_READLINE=1 -DENABLED_LOCAL_INFILE=1 -  
DMYSQL_DATADIR=/data/mysql -DMYSQL_USER=mysql
```

( 简化 : cmake

```
-DCMAKE_INSTALL_PREFIX=/usr/local/mysql
```

```
-DSYSCONFDIR=/etc
```

```
-DDEFAULT_CHARSET=utf8
```

```
-DDEFAULT_COLLATION=utf8_general_ci
```

```
-DWITH_EXTRA_CHARSETS=all )
```

```
# make
```

```
# make install
```

```
# chmod +w /usr/local/mysql
```

```
# chown -R mysql:mysql /usr/local/mysql
```

```
# ln -s /usr/local/mysql/lib/libmysqlclient.so.16 /usr/lib/libmysqlclient.so.16
```

```
# mkdir -p /var/mysql/
```

```
# mkdir -p /var/mysql/data/
```

```
# mkdir -p /var/mysql/log/
```

```
# chown -R mysql:mysql /var/mysql/
```

```
# cd support-files/
```

```
# cp my-large.cnf /etc/my.cnf (注意:my-large.cnf适用于1G内存左右的服务器，可以根据自己配置情况  
选用my-large.cnf 或 my-huge.cnf 等不同配置)
```

```
# cp mysql.server /etc/init.d/mysqld
```

#### 四. 配置启动MySQL 5.5.22

1. 若有需要请先修改 mysql 的配置 my.cnf

```
# vi /etc/my.cnf
```

2. mysql 初始化安装

```
# /usr/local/mysql/scripts/mysql_install_db \
```

```
--defaults-file=/var/mysql/my.cnf \ ( 应该是/etc/my.cnf )
```

```
--basedir=/usr/local/mysql \
```

```
--datadir=/var/mysql/data \
```

```
--user=mysql
```

### 3. 将 mysql 加入开机启动

```
# chmod +x /etc/init.d/mysqld
# vi /etc/init.d/mysqld ( 编辑此文件 , 查找并修改以下变量内容 : )
    basedir=/usr/local/mysql
    datadir=/var/mysql/data

# chkconfig --add mysqld
# chkconfig --level 345 mysqld on
```

### 4. 启动 mysql

```
# service mysqld start

# echo "PATH=$PATH:/usr/local/mysql/bin" >> /etc/profile
# . /etc/profile \\立即生效

# netstat -anput | grep mysqld    \\查看端口

# service mysqld status    \\查看服务状态
```

-----访问MySQL-----

远程访问 :

```
# mysql -u 用户名 -h ip地址 -p 密码
```

修改密码 :

```
# mysqladmin -u root password "redhat"
```

RHEL6自带客户端访问工具mysql, 如果没有需要安装mysql

```
CREATE TABLE users(user_name CHAR(16) NOT NULL,user_passwd CHAR(48) DEFAULT '',PRIMARY
KEY (user_name));
```

mysql5.5备份出错

```
mysqldump: Got error: 1142: SELECT,LOCK TABLE command denied to user 'root'@'localhost' for
table 'cond_instances' when using LOCK TABLES
```

原因 : 在mysql5.5中, 增加了performance\_schema 导致

解决方案

```
mysqldump -u root -p --skip-lock-tables performance_schema > performance_schema.sql
```

显示错误信息#perror 113 ( 错误报错代码 )

MySQL主辅复制

## 配置mysql5.5主从服务器

教程开始：

### 一、安装MySQL

说明：在两台MySQL服务器192.168.21.169和192.168.21.168上分别进行如下操作，安装MySQL 5.5.22

### 二、配置MySQL主服务器 ( 192.168.21.169 )

```
mysql -uroot -p #进入MySQL控制台
```

```
create database handb; #建立数据库handb
```

```
insert into mysql.user(Host,User,Password) values('localhost','bakuser',password('123456'));
```

```
#创建用户bakuser
```

#授权用户bakuser只能从192.168.21.168这个IP访问主服务器192.168.21.169上面的数据库，并且只具有数据库备份的权限

```
grant replication slave on *.* to 'bakuser'@'192.168.21.168' identified by '123456' with grant option;
```

```
flush privileges; #刷新系统授权表
```

### 三、把MySQL主服务器192.168.21.169中的数据库handb导入到MySQL从服务器192.168.21.168中

#### 1、导出数据库handb

```
mysqldump -u root -p handb > /home/handbbak.sql
```

#在MySQL主服务器进行操作，导出数据库handb到/home/handbbak.sql

备注：在导出之前可以先进入MySQL控制台执行下面命令

```
flush tables with read lock; #数据库只读锁定命令，防止导出数据库的时候有数据写入
```

```
unlock tables; #解除锁定
```

#### 2、导入数据库到MySQL从服务器

```
mysql -u root -p #进入从服务器MySQL控制台
```

```
>create database handb; #创建数据库
```

```
>use handb #进入数据库
```

( scp handbbak.sql [root@192.168.56.201](mailto:root@192.168.56.201):/root/ #将数据库备份文件拷贝到从数据库。 )

>source /home/handbbak.sql #导入备份文件到数据库

mysql -u bakuser -h 192.168.21.169 -p #测试在从服务器上登录到主服务器

#### 四、配置MySQL主服务器的my.cnf文件

vi /etc/my.cnf #编辑配置文件，在[mysqld]部分添加下面内容  
server-id=1 #设置服务器id，为1表示主服务器，注意：如果原来的配置文件中已经有这一行，就不用再添加了。  
log-bin=mysql-bin #启动MySQL二进制日志系统，注意：如果原来的配置文件中已经有这一行，就不用再添加了。  
binlog-do-db=handb #需要同步的数据库名，如果有多个数据库，可重复此参数，每个数据库一行  
binlog-ignore-db=mysql #不同步mysql系统数据库  
service mysqld restart #重启MySQL

mysql -u root -p #进入mysql控制台  
show master status; #查看主服务器，出现以下类似信息  
+-----+-----+-----+-----+  
| File | Position | Binlog\_Do\_DB | Binlog\_Ignore\_DB |  
+-----+-----+-----+-----+  
| mysql-bin.000019 | 7131 | osyunweidb | mysql |  
+-----+-----+-----+-----+  
1 row in set (0.00 sec)

**注意：这里记住File的值：mysql-bin.000019和Position的值：7131，后面会用到。**

#### 五、配置MySQL从服务器的my.cnf文件

vi /etc/my.cnf #编辑配置文件，在[mysqld]部分添加下面内容  
server-id=2 #配置文件中已经有一行server-id=1，修改其值为2，表示为从数据库  
log-bin=mysql-bin #启动MySQL二进制日志系统，注意：如果原来的配置文件中已经有这一行，就不用再添加了。  
replicate-do-db=handb #需要同步的数据库名，如果有多个数据库，可重复此参数，每个数据库一行  
replicate-ignore-db=mysql #不同步mysql系统数据库

service mysqld restart #重启MySQL

**注意：MySQL 5.1.7版本之后，已经不支持把master配置属性写入my.cnf配置文件中了，只需要把同步的数据库和要忽略的数据库写入即可。**

mysql -u root -p #进入MySQL控制台  
slave stop; #停止slave同步进程  
change master to

master\_host='192.168.21.169',

master\_user='bakuser',

master\_password='123456',

master\_log\_file='mysql-bin.000019',

master\_log\_pos=7131; #执行同步语句  
slave start; #开启slave同步进程  
SHOW SLAVE STATUS\G #查看slave同步信息，出现以下内容

\*\*\*\*\* 1. row \*\*\*\*\*

Slave\_IO\_State: Waiting for master to send event  
Master\_Host: 192.168.21.169  
Master\_User: osyunweidbbak  
Master\_Port: 3306  
Connect\_Retry: 60  
Master\_Log\_File: mysql-bin.000019

```
Read_Master_Log_Pos: 7131
Relay_Log_File: MySQLSlave-relay-bin.000002
Relay_Log_Pos: 253
Relay_Master_Log_File: mysql-bin.000019
Slave_IO_Running: Yes
Slave_SQL_Running: Yes
Replicate_Do_DB: osyunweidb
Replicate_Ignore_DB: mysql
Replicate_Do_Table:
Replicate_Ignore_Table:
1 row in set (0.00 sec)
```

**注意查看：**

**Slave\_IO\_Running: Yes**

**Slave\_SQL\_Running: Yes**

**以上这两个参数的值为Yes，即说明配置成功！**

在主mysql上测试

mysql >show processlist;//显示从服务器上的进程

## 六、测试MySQL主从服务器双机热备是否成功

### 1、进入MySQL主服务器

mysql -u root -p #进入主服务器MySQL控制台

use handb #进入数据库

CREATE TABLE test ( id int not null primary key,name char(20) ); #创建test表

### 2、进入MySQL从服务器

mysql -u root -p #进入MySQL控制台

use osyunweidb #进入数据库

show tables; #查看handb表结构，会看到有一个新建的表test，表示数据库同步成功

**至此，MySQL数据库配置主从服务器实现双机热备实例教程完成**

## MySQL数据同步（一主多从）

MySQL数据同步一主多从和一主一从原理其实是一样。

环境:

Master centos5.2 mysql5.1.35 主服务器的ip : 10.207.10.154

Slave1 centos5.2 mysql5.6.10 从服务器的ip : 10.207.10.158

Slave2 centos5.2 mysql5.6.10 从服务器的ip : 10.207.10.205

—master主服务器上设置：

### 1、编辑my.ini文件

server-id=1 //添加server-id必须为1到232-1之间的一个正整数

log-bin = /var/log/mysql/logbin/mysql-bin //开启log-bin二进制日志文件(注意logbin的权限必须对

mysql是可写的)

binlog-do-db = yaxin //同步yaxin库 其他库都不同步

#binlog-ignore-db = mysql,information\_schema //不同步

2、重启mysql数据库

3、添加slave服务器访问master服务器的帐号和权限

mysql>GRANT ALL PRIVILEGES ON \*.\* TO 'user'@'10.207.10.158' IDENTIFIED BY '123456' ;

mysql>GRANT ALL PRIVILEGES ON \*.\* TO 'user'@'10.207.10.205' IDENTIFIED BY 'abcedf' ;

mysql> show master status; //得到主服务器上当前的二进制日志名和偏移量

```
+-----+-----+-----+-----+
| File           | Position | Binlog_Do_DB | Binlog_Ignore_DB | +-----+-----+-----+
+-----+-----+
| mysql-bin.000006 | 106 | yaxin       |                   | +-----+-----+-----+
```

1 row in set

二slave1从服务器上设置：

1、编辑my.cnf文件(vi /etc/my.cnf)

server-id=2 slave的ID号，此处一定要大于master端。

replicate-do-db = yaxin

2、重启mysql数据库

3、同步

mysql>stop slave;

mysql>change master to

>master\_host='10.207.10.154',

>master\_user='myuser', master端创建的用于主从同步的账户和密码

>master\_password='abc123',

>master\_log\_file='mysql-bin.000006', //master端记录的file值

>master\_log\_pos=106; // master端记录的position值

mysql>start slave;

mysql>show slave status \G

mysql >show processlist; //显示从服务器上的进程

三slave2从服务器上设置：

1、编辑my.cnf文件(vi /etc/my.cnf)

server-id=3

replicate-do-db = yaxin

2、重启mysql数据库

3、同步

mysql>stop slave;

mysql>change master to

>master\_host='10.207.10.154',

>master\_user='myuser', master端创建的用于主从同步的账户和密码

>master\_password='abc123',

>master\_log\_file='mysql-bin.000006', //master端记录的file值

>master\_log\_pos=106; // master端记录的position值

mysql>start slave;

mysql>show slave status \G

mysql >show processlist; //显示从服务器上的进程

四、测试

注意：主从数据库初始信息要相同，包括表结构、表数据、编码等，如果不相同，则将master主服务器中的信息先备份，然后再导入到从服务器。

