# mysql学习随笔

测试环境

Linux系统

mysql 5.5.12

客户端 SSH

# 一、安装mysql

采用已编译好的二进制文件压缩包安装，版本mysql-5.5.12-linux2.6-x86\_64

1. 创建一个路径，并拷贝文件到指定路径下

cd / #选择根目录

mkdir data #在根目录下建立目录data

cd /data #进入data目录

mkdir mysql #建立目录mysql

假设压缩包在/opt/software路径下，将压缩包拷贝到/data路径下

cp /opt/software/mysql-5.0.41-linux-i686-glibc23.tar /data/mysql

如果压缩抱在其他linux服务器上的/opt/software路径下

scp root@12.168.1.100:/opt/software/ mysql-5.0.41-linux-i686-glibc23.tar /data/mysql

回车后输入root密码即可。

1. 解压缩

cd /data/mysql

tar xvf mysql-5.0.41-linux-i686-glibc23

3、为解压缩后的路径改名

mv mysql-5.0.41-linux-i686-glibc23 mysql

4、移动mysql安装文件到/usr/local路径下

mv /data/mysql/mysql /usr/local

5、建立存放mysql运行文件的目录

cd /data/mysql

mkdir run #运行目录

mkdir innodb #innodb的存储目录

mkdir binlog #binlog的存储目录

mkdir data #系统数据库和用户数据库文件的存储目录

6、创建mysql运行账户并修改mysql文件所有者权限

此时，由于第4步把安装文件移动到了/usr/local路径下

1. 创建mysql用户组

groupadd mysql

1. 创建用户，并指定用户在mysql用户组

useradd –g mysql mysql

1. 修改mysql文件所有者权限

chown –R mysql:mysql /data/mysql

chown –R mysql:mysql /usr/local/mysql

7、复制mysql配置文件到/etc目录下，/usr/local/mysql/support-files目录下有4个配置文件（my-huge.cnf（巨大），my-large.cnf（大），my-medium.cnf（一般），my-small.cnf（小），分别指mysql的数据库容量）根据需求复制对应的配置文件到/etc路径下。

cp /usr/local/mysql/support-files/my-medium.cnf /etc/my.cnf

8、安装mysql(这一步其实是初始化mysql)

/usr/local/mysql/scripts/mysql\_install\_db --basedir=/usr/local/mysql --datadir=/data/mysql/image/data --user=mysql

9、启动mysql

/usr/local/mysql1/bin/mysqld\_multi  --defaults-file=/usr/local/my.cnf  start 1

查看mysql是否启动

/usr/local/mysql1/bin/mysqld\_multi report

为启动配置文件建立链接

ln -s /data/mysql /run/mysql.socket /tmp/mysql.sock

10、设置mysql的root账户

/usr/local/mysql/bin/mysqladmin -u root password 123456

本地 连接到mysql，允许远程连接mysql

[root@localhost bin]# mysql –uroot –p

回车输入密码

mysql>GRANT ALL PRIVILEGES ON \*.\* TO 'root'@'%' IDENTIFIED BY  '12345678' WITH GRANT OPTION;

刷新配置，使得配置生效

flush privileges;

# 安装问题：

1. 第7步，由于mysql的配置文件内容是注释过的，需要去掉需要配置的每行的最前面的#号。

这里说明一下配置文件，熟悉配置文件对mysql的性能和高可用性有很大帮助。

基本介绍：

1. []表示节点，这个类似与xml文件中的一个字节点，直到下一个[]之间都是第1个[]节点配置属性。
2. #号代表mysql的注释

节点介绍：

1. [client]客户端节点

#mysql服务器启动必须的端口号，这个端口号是可以修改的，但是一定是系统还没被使用过的端口号。

port = 3306

#socket文件路径，socket文件是unix下进程间通信用的，mysql.sock就是客户端连接与mysql间通信用的socket文件，只能本机使用，远程连接要通过tcp/ip了

socket = /tmp/mysql.sock

1. [mysqld\_multi]节点，该节点是配置mysql的相关启动文件路径

#mysql启动文件的路径，mysqld和mysqld\_safe都是启动mysql的文件，但实际上在linux系统中，都是推荐使用mysqld\_safe这个文件启动mysql。这里的配置告诉系统，启动mysql的文件路径。

mysqld = /usr/local/mysql/bin/mysqld\_safe

#mysqladmin ，在安装mysql后，运行这个文件可以设置账户root的密码，同时它也可以停止mysql服务运行。

mysqladmin = /usr/local/mysql/bin/mysqladmin

#超级管理员账户和密码。实际上，即使在配置文件中手动改变password的值，密码也无法生效，必须通过mysqladmin改变密码后，密码才能生效。

user = root

password = 123456

1. [mysqld1]，本机mysql实例，注意：如果只安装了一个实例，会只有一个[mysqld1]，安装多个实例，只需再配置一个[mysqld2]的节点，这里为了区别，这里使用[mysqld1]，实际上，一个实例的mysql，节点应该是[mysqld]。

#这里配置的是mysql的运行账户。

user=mysql

#运行当前实例的mysql的端口号，这个跟客户端的端口号一样可以修改。

port = 3306

#socket文件路径，socket文件是unix下进程间通信用的，mysql.sock就是客户端连接与mysql间通信用的socket文件，只能本机使用，远程连接要通过tcp/ip了

socket = /data/mysql/run/mysql.socket

#启动mysql进程的系统pid

pid-file = /data/mysql/run/mysql.pid

#mysql数据文件的存储路径

datadir = /data/mysql/data

#mysql普通日志文件路径

general\_log\_file = /data/mysql/mysql-access.log

#mysql的错误日志路径

log-error = /data/mysql/mysql-error.log

#主从复制时，binlog文件的过期时间，如果没有做主从，可以不配置

expire\_logs\_days=3

# key\_buffer是用于索引块的缓冲区大小，增加它可得到更好处理的索引(对所有读和多重写)。  
 索引块是缓冲的并且被所有的线程共享，key\_buffer的大小视内存大小而定。

key\_buffer = 2048M

#为所有线程打开表的数量。增加该值能增加mysqld要求的文件描述符的数量。可以避免频繁的打开数据表产生的开销

table\_cache = 10240

#每个需要进行排序的线程分配该大小的一个缓冲区。增加这值加速ORDER BY或GROUP BY操作。  
                               注意：该参数对应的分配内存是每连接独占！如果有100个连接，那么实际分配的总共排序缓冲区大小为100×40=4000MB

sort\_buffer\_size = 40M

#读查询操作所能使用的缓冲区大小。和sort\_buffer\_size一样，该参数对应的分配内存也是每连接独享。

read\_buffer\_size = 40M

#改参数在使用行指针排序之后，随机读用的

read\_rnd\_buffer\_size = 80M

#MyISAM表发生变化时重新排序所需的缓冲

myisam\_sort\_buffer\_size = 512M

#缓存可重用的线程数

thread\_cache=64

#服务器最大连接数

max\_connections=1000

#用户最大连接数

max\_user\_connections=1000

#最大信息包大小

max\_allowed\_packet=10G

#TCP/IP和套接字通信缓冲区大小

net\_buffer\_length=10M

#指定MySQL查询结果缓冲区的大小

query\_cache\_size=512M

#最大并发线程数，取值为服务器逻辑CPU数量×2，如果CPU支持H.T超线程，再×2

thread\_concurrency=48

#禁止MySQL对外部连接进行DNS解析，使用这一选项可以消除MySQL进行DNS解析的时间。  
 但需要注意，如果开启该选项，则所有远程主机连接授权都要使用IP地址方式，否则MySQL将无法正常处理连接请求！

skip-name-resolve

#区分大小写

lower\_case\_table\_names=0

#服务器字符集编码

character\_set\_server = utf8

#制定编码

init\_connect='SET NAMES utf8'

#关闭该选项可能会在安装时报错

skip-federated

#记录执行慢的语句的文件

log-slow-queries=slow\_query.txt

#执行慢的语句的阀值，该处为1秒

long\_query\_time=1

#做主从时，主master配置

log-bin=master-bin #binlog的文件名

log-bin-index=master-bin.index #binlog的索引位置，记录binlog文件顺序

server-id=1 #主mysql服务器ID，必须唯一

#做主从时，从slave配置

relay-log=slave-relay-bin #binlog文件名

relay-log-index=slave-relay-bin.index #binlog的索引位置，记录binlog文件顺序

server-id=2 #从mysql服务器ID,必须唯一

#做主从时，binlog文件的复制模式，该处为混合复制模式(binlog复制模式STATEMENT(基于语句复制)，ROW(基于行复制)，MIXED(混合复制))。

binlog\_format=mixed

#主从运行时，当主宕机，可以让从升级为主

log\_slave\_updates=1

#复制时，忽略系统库mysql的所有对象

replicate-wild-ignore-table=mysql.%

#复制时，忽略跨库更新information\_schema，performance\_schema下的所有对象

replicate-wild-ignore-table=information\_schema.%

replicate-wild-ignore-table=performance\_schema.%

# InnoDB表的目录共用设置

innodb\_data\_home\_dir = /data/mysql/innodb/

#innodb文件名：文件大小：增长方式。当存储空间不足时，最后一个数据文件会自动增长，每次增长8MB。

innodb\_data\_file\_path = ibdata1:512M:autoextend

#此参数确定日志文件组中的文件的位置，日志组中文件的个数由innodb\_log\_files\_in\_group确定，此位置设置默认为MySQL的datadir

innodb\_log\_group\_home\_dir = /data/mysql/innodb/

#可以修改InnoDB为独立表空间模式，每个数据库的每个表都会生成一个数据空间

innodb\_file\_per\_table = 1

#innodb缓冲池大小

innodb\_buffer\_pool\_size = 409M

#除了缓存表数据和索引外,为操作所需的其他内部项分配缓存

innodb\_additional\_mem\_pool\_size = 10M

#innodb日志文件大小

innodb\_log\_file\_size = 32M

#innodb日志缓冲池大小

innodb\_log\_buffer\_size = 8M

#innodb日志刷新到磁盘的时间，0为每隔1秒刷新一次，2为每次提交事务日志时刷新到磁盘，1为事务日志将在每次提交后刷新到磁盘。

innodb\_flush\_log\_at\_trx\_commit = 0

#innodb锁的等待时间，防止出现长时间锁定而形成阻塞。

innodb\_lock\_wait\_timeout = 50

1. [mysqldump]节点

#快速导出数据

quick

#导出数据时，最大数据包大小

max\_allowed\_packet = 16M

1. [mysql]节点

#自动补全功能，即输入前几位字符后按tab键带出完整的字符串

no-auto-rehash

1. [isamchk]存储引擎设置节点

#索引缓存大小

key\_buffer = 20M

#排序缓存大小

sort\_buffer\_size = 20M

#每个独立读取缓存大小

read\_buffer = 2M

#每个独立写入缓存大小

write\_buffer = 2M

1. [myisamchk]存储引擎节点

#索引缓存大小

key\_buffer = 20M

#排序缓存大小

sort\_buffer\_size = 20M

#每个独立读取缓存大小

read\_buffer = 2M

#每个独立写入缓存大小

write\_buffer = 2M

1. [mysqlhotcopy]

#服务器关闭交互式连接前等待活动的秒数，默认为28800秒

interactive-timeout

1. 第9步时，无法启动mysql服务。

原因：

1. 端口被占
2. 用同样的mysql服务已经启动。
3. 查看mysql的日志，路径在my.cnf配置文件中的log-error = /data/mysql/mysql-error.log，
4. 报错Can't find command 'my\_print\_defaults'。

则是因为/etc/profile配置被重置，执行

[root@localhost bin]# echo "PATH=/usr/local/mysql/bin:\$PATH" >> /etc/profile

[root@localhost bin]# source /etc/profile

1. 报错 ‘/tmp/mysql.sock’ is not exists。可能是运行账户的权限问题，也可能是端口被占，或mysql进程已启用。

3、第10步，mysql没有密码是不允许访问，但是在mysql的客户端navicat是可以连接的。所以必须给mysql设置超级管理员密码。

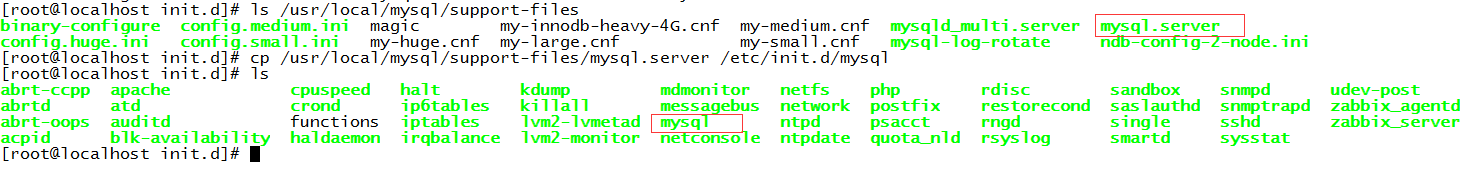
## 开机自动启动mysql

1. 修改/etc/my.cnf，使得红框部分的socket值相同，并且路径都是/tmp/mysql.sock，因为这是系统默认的文件路径。

注意：如果是多实例，则无法使用service mysql 来启动。此外在服务器只有一个mysql实例时，my.cnf的[mysqld]节点必须是[mysqld]，而不是[mysqld1],[mysqld2]等等。

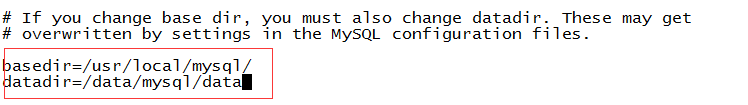


1. 复制mysql安装路径下的mysql.server文件到/etc/init.d路径下。并将mysql文件所有者赋予运行mysql服务的账户。

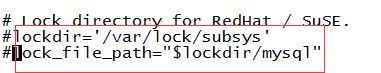




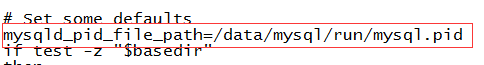
3、修改mysql文件，找到以下节点，将basedir的值修改为mysql的安装目录，将datadir的值修改为数据存放目录。

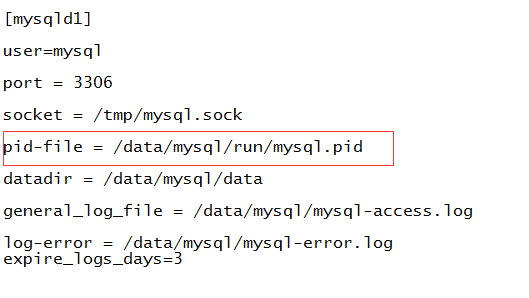


注释以下节点

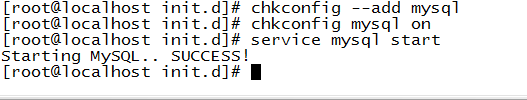


将以下节点的值，改为/etc/my.cnf配置文件中pid-file的值





执行以下命令：



重启服务器，测试是否开机启动

# 二、mysql备份还原

备份数据库

mysqldump -uroot -p123456 studentinfo>>/test/studentinfo\_bak.sql

还原数据库，还原时，数据库必须有一个名字相同的数据库，否则会报错没有目标数据库

mysql -uroot -p123456 studentinfo </test/studentinfo\_bak.sql

使用binlog文件还原数据库

1. 先还原最近一个备份文件到数据库

mysql -uroot -p123456 studentinfo </test/studentinfo\_bak.sql

1. 再还原binlog

mysqlbinlog --no-defaults mysql-bin.000007 --start-pos=382774862 --stop-datetime='2014-10-23 14:30:00' --database= studentinfo |mysql -uroot -p123456 –f

1. mysql-bin.000007是binlog文件，可以写多个，但是必须用空格分隔
2. --start-pos=382774862是指从binlog的哪个位置开始还原，这里可以是--start-datetime='2014-10-23 12:00:00'指从哪个时间点开始还原。
3. --stop-datetime='2014-10-23 14:30:00'指还原到哪里结束。
4. --database= studentinfo指定还原的数据库
5. |mysql -uroot -p123456使用还原的账户密码
6. –f跳过主键冲突

# 三、mysql查询

已存在root账户并设立了密码，连接mysql

[root@localhost bin]# mysql -u root -p

Enter password:

错误集锦：

1．错误提示：Can't connect to local MySQL server through socket '/tmp/mysql.sock' (2)

原因分析：

由于服务器重启，导致tmp/mysql.sock文件被删除，需要把my.cnf文件中

socket = /data/mysql/image/run/mysql.socket这个节点的值重新复制一份到/tmp路径下，mysql.sock

执行以下语句：

ln -s /data/mysql/image/run/mysql.socket /tmp/mysql.sock

1. 创建账户

mysql> insert into mysql.user(host,user,password)

-> values('localhost','testuser',password('123'));

2、删除账户

mysql> delete from mysql.user where user='testuser';

3、修改账户密码

mysql>update mysql.user set password=password(‘1234’) where user=’testuser’;

4、查询用户信息

mysql> select host,user,password from mysql.user;

以上三点的操作可以归纳为：

1. mysql.user可以看做一个系统的用户表，这个表里面记录着用户登录信息。
2. 创建用户相当于在mysql.user表中新增一条记录。
3. 更新用户信息相当于更新mysql.user表的信息。
4. 删除用户相当于删除mysql.user表中的信息。
5. 可以通过查询mysql.user表来查看用户信息。

5、赋予权限

（1）授予访问权限，执行以下语句，grant为赋予权限的命令；all privileges代表所有权限；\*.\*，\*为通配符，第一个\*代表所有数据库，第二个\*代表所有表；user1代表用户名，’%’代表所有IP或主机名 ‘123’代表访问密码；flush privileges 刷新配置，使得配置生效。

mysql> grant all privileges on \*.\* to user1@'%' identified by '123';

mysql>flush privileges;

（2）授予数据库权限

mysql> grant select,insert,update,delete on testmysql.st\_class to 'testuser'@'localhost' identified by '123' with grant option;

select,insert,update,delete分别对应查询，插入，更新，删除4中权限，testmysql.st\_class，其中testmysql表示数据库名，st\_class表示表名。

注意：在给予某个用户特定的数据库的特定权限时，一定要在该数据库下才能赋予。

select,insert,update,delete可以被 all代替，即赋予所有权限。testmysql.st\_class可以被\*.\*代替，即针对所有数据库。

执行以下语句时，不需要在特定的数据库下。

mysql> grant ALL PRIVILEGES on \*.\* to 'testuser'@'localhost' identified by '123' with grant option;

（3）赋予可以远程访问testmysql数据库的账户

mysql> grant all privileges on testmysql.\* to 'testuser'@'%' identified by '1234';

6、创建表

Create table tb\_test

(

Id int auto\_increment primary key not null,

Name varchar(10) not null

)

创建带有联合主键的表

Create table tb\_test

(

Id int not null,

Tid int not null,

Name varchar(10) not null,

Primary key (id,tid)

)

修改字段长度

alter table test modify column id smallint not null;

修改字段名

alter table test change name tname varchar(10) not null;

增加字段

alter table test add column remark varchar(200) null;

删除字段

alter table test drop column remark;

创建普通索引（类似与MSSQL的非聚集索引）

create index ix\_tname on test(tname);

创建唯一索引

create unique index uq\_tid on test(tid);

删除主键

alter table test drop primary key;

创建主键

alter table test add primary key(id,tid);

删除索引

alter table test drop index uq\_tid;

注意：

使用alter table 可以创建或删除主键，删除普通和唯一索引。

直接使用create 只可以创建普通索引和唯一索引。

删除主键时，如果没有创建PRIMARY KEY索引，但表具有一个或多个UNIQUE索引，则MySQL将删除第一个UNIQUE索引。

查看数据库

Mysql> show databases;

查看表

Mysql>show tables;

查看字段

Mysql>describe tableName;

注意：

数据类型为Bit的字段，值是无法直接显示的。

# csv文件导入mysql

mysql> load data infile '/liling\_test/test\_files/slow\_query\_20141022.csv' into table slowquery1 fields terminated by '|' lines terminated by '\n';

# mysql导出文本文件

select \* from slowquery into outfile '/liling\_test/test\_files/slowquery.txt' fields terminated by '|' OPTIONALLY ENCLOSED BY '"' lines terminated by '\n';

如果报错：

ERROR 1 (HY000): Can't create/write to file '/liling\_test/test\_files/slowquery.txt' (Errcode: 13)

说明：mysql没有权限在/liling\_test/test\_files/创建文件slowquery.txt。需要指定路径到mysql的启动账户的路径下，即：/home/mysql/，其中mysql是指启动mysql服务器的linux账户。

# mysql查询数据库空间大小

SELECT CONCAT(TRUNCATE(SUM(data\_length)/1024/1024,2),'MB') AS data\_size,

CONCAT(TRUNCATE(SUM(max\_data\_length)/1024/1024,2),'MB') AS max\_data\_size,

CONCAT(TRUNCATE(SUM(data\_free)/1024/1024,2),'MB') AS data\_free,

CONCAT(TRUNCATE(SUM(index\_length)/1024/1024,2),'MB') AS index\_size

FROM information\_schema.tables WHERE TABLE\_SCHEMA = 'hofan\_db\_monitor';

# mysql查询表空间大小

SELECT CONCAT(TRUNCATE(SUM(data\_length)/1024/1024,2),'MB') AS data\_size,

CONCAT(TRUNCATE(SUM(max\_data\_length)/1024/1024,2),'MB') AS max\_data\_size,

CONCAT(TRUNCATE(SUM(data\_free)/1024/1024,2),'MB') AS data\_free,

CONCAT(TRUNCATE(SUM(index\_length)/1024/1024,2),'MB') AS index\_size

FROM information\_schema.tables WHERE TABLE\_NAME = 'amazon\_detail';