# 实战 centos6.5下mysql-5.6.19数据库安装、优化、互为主从 mysql集群 高可用

## 一．环境介绍

mysql互为主从环境介绍：

系统：centos6.5\_64\_mini

mysql版本：mysql-5.6.19

实验节点如下：

节点一：172.16.20.195

节点二：172.16.20.195

## 二．安装环境及卸载自带mysql

首先安装需要用到的库并删除自带mysql （节点一和二）

# yum -y install gcc gcc-c++ ncurses ncurses-devel openssl openssl-devel cmake perl lsof wget bison

# rpm -qa|grep mysql

# rpm -e --allmatches --nodeps mysql-libs-5.1.71-1.el6.x86\_64

## 三．安装mysql

MYSQL的安装（节点一和二）

### 2.1创建MySQL数据库存放目录、配置用户和用户组

# /usr/sbin/groupadd mysql

# /usr/sbin/useradd -s /sbin/nologin -M -g mysql mysql

# mkdir -p /home/softisland/mysqldata

# chown -R mysql:mysql /home/softisland/mysqldata

### 2.2创建软件下载目录，把所需软件下载

# mkdir /home/softisland/

# cd /home/softisland/

# wget http://down.yunwei8.com/soft/linux/mysql-5.6.19.tar.gz

# wget http://down.yunwei8.com/soft/linux/libunwind-1.1.tar.gz

# wget http://down.yunwei8.com/soft/linux/gperftools-2.2.tar.gz

### 2.3安装mysql-5.6.19

# tar zxvf mysql-5.6.19.tar.gz

# cd mysql-5.6.19

# cmake -DCMAKE\_INSTALL\_PREFIX=/home/softisland/mysql5.6 -DMYSQL\_DATADIR= /home/softisland/mysqldata -DWITH\_INNOBASE\_STORAGE\_ENGINE=1 -DWITH\_MEMORY\_STORAGE\_ENGINE=1 -DWITH\_MYISAM\_STORAGE\_ENGINE=1 -DSYSCONFDIR=/etc/ -DWITH\_SSL=yes -DDEFAULT\_CHARSET=utf8 -DDEFAULT\_COLLATION=utf8\_general\_ci -DWITH\_READLINE=on

# make

# make install

# chmod +w /home/softisland/mysql5.6

# chown -R mysql:mysql /home/softisland/mysql5.6

### 2.4创建软连接

# ln -s /home/softisland/mysql5.6/lib/lib\* /usr/lib/

### 2.5创建my.cnf配置文件，并修改

设置Mysql

# cp /home/softisland/ mysql-5.6.19/support-files/my-default.cnf /etc/my.cnf

# vi /etc/my.cnf

在[mysqld]下添如下行

character-set-server = utf8

default-storage-engine = MyISAM

basedir = /home/softisland/mysql5.6

datadir = /home/softisland/mysqldata

log-error = /home/softisland/mysql5.6/mysql\_error.log

pid-file = /home/softisland/mysql5.6/mysql.pid

max\_allowed\_packet = 32M

explicit\_defaults\_for\_timestamp = true

### 2.6以mysql用户帐号的身份建立数据表

# /home/softisland/mysql5.6/scripts/mysql\_install\_db --basedir=/home/softisland/mysql5.6--datadir=/home/softisland/mysqldata --user=mysql

### 2.7设置mysql开机自动启动服务

# cp /home/softisland/mysqldata/support-files/mysql.server /etc/init.d/mysqld

# chkconfig --add mysqld

# chkconfig --level 345 mysqld on

### 2.8修改服务配置文件

# vi /etc/init.d/mysqld

根据设定需要，修改mysqld文件中的下面两项

basedir=/home/softisland/mysql5.6

datadir=/home/softisland/mysqldata

### 2.9启动mysqld服务

# service mysqld start

### 2.10登录mysql

通过命令行登录管理MySQL服务器并设置root密码（提示输入密码时直接回车）：（节点一和二）

# /home/softisland/mysql5.6/bin/mysql -u root -p -S /tmp/mysql.sock

# mysql> SET PASSWORD FOR 'root'@'localhost' = PASSWORD('000000');

退出mysql

# mysql> exit

### 2.11软连接mysql

# ln -s /home/softisland/mysql5.6/bin/mysql /usr/bin

### 2.12用root用户连接mysql

尝试用root连接mysql

# mysql -u root -p

成功登录后查看状态

# mysql> status;

## 四．Mysql优化

mysql优化（节点一和二）

### 4.1安装libunwind

为MySQL添加TCMalloc库的安装步骤（Linux环境）：

1. TCMalloc（Thread-Caching Malloc）是google开发的开源工具──“google-perftools”中的成员。与标准的glibc库的malloc相比，TCMalloc在内存的分配上效率和速度要高得多，可以在很大程度上提高MySQL服务器在高并发情况下的性能，降低系统负载。
2. 64位操作系统请先安装libunwind库，32位操作系统不要安装。libunwind库为基于64位CPU和操作系统的程序提供了基本的堆栈辗转开解功能，其中包括用于输出堆栈跟踪的API、用于以编程方式辗转开解堆栈的API以及支持C++异常处理机制的API。

首先安装libunwind-1.1

# cd /home/softisland

# tar zxvf libunwind-1.1.tar.gz

# cd libunwind-1.1

# CFLAGS=-fPIC ./configure

# make CFLAGS=-fPIC

# make CFLAGS=-fPIC install

### 4.2安装google-perftools

# cd /home/softisland

# tar zxvf gperftools-2.2.tar.gz

# cd gperftools-2.2/

# ./configure

# make && make install

# echo "/usr/local/lib" > /etc/ld.so.conf.d/usr\_local\_lib.conf

# /sbin/ldconfig

### 4.3修改MySQL启动脚本

修改MySQL启动脚本（根据你的MySQL安装位置而定）

# vi /home/softisland/mysql5.6/bin/mysqld\_safe

在# executing mysqld\_safe的下一行，加上：

export LD\_PRELOAD=/usr/local/lib/libtcmalloc.so

保存后退出，然后重启MySQL服务器。

# service mysqld restart

### 4.4查看tcmalloc是否生效

使用lsof命令查看tcmalloc是否起效：

# /usr/sbin/lsof -n | grep tcmalloc

如果发现以下信息，说明tcmalloc已经起效：

mysqld 11227 mysql mem REG 8,1 2171746 18287 /usr/local/lib/libtcmalloc.so.4.1.2

## 五．设置mysql互为主备

mysql互为主从配置

### 5.1防火墙开启3306端口

第一步防火墙设置（节点一和二）

端口开放

# /sbin/iptables -I INPUT -p tcp --dport 3306 -j ACCEPT

然后保存：

# /etc/rc.d/init.d/iptables save

### 5.2配置mysql互为主从

#### 5.2.1登录mysql数据库（节点一和二）

# mysql-u root -p

#### 5.2.2建立需要同步的数据库,同时考虑分片（节点一和二）

# mysql> create database softisland;

# mysql> create database softisland1;

# mysql> create database softisland2;

#### 5.2.3添加远程用户

添加一个远程用户，名为softisland密码为123456,并且只能对softisland数据库有权限（节点一和二）

# mysql> GRANT ALL PRIVILEGES ON softisland.\* TO softisland@"%" IDENTIFIED BY '123456';

#### 5.2.4修改/etc/my.cnf文件（节点一）

# rm -rf /etc/my.cnf

# vi /etc/my.cnf

添加如下配置代码

[client]

default-character-set = utf8

port = 3306

socket = /tmp/mysql.sock

[mysqld]

character-set-server = utf8

default-storage-engine = MyISAM

basedir = /home/softisland/mysql5.6

datadir = /home/softisland/mysqldata

log-error = /home/softisland/mysql5.6/mysql\_error.log

pid-file = /home/softisland/mysql5.6/mysql.pid

port = 3306

socket = /tmp/mysql.sock

skip-external-locking

key\_buffer\_size = 384M

max\_allowed\_packet = 32M

table\_open\_cache = 512

sort\_buffer\_size = 2M

read\_buffer\_size = 2M

read\_rnd\_buffer\_size = 8M

myisam\_sort\_buffer\_size = 64M

thread\_cache\_size = 8

query\_cache\_size = 32M

thread\_concurrency = 8

log-bin=mysql-bin

server-id = 1

binlog-do-db= softisland

replicate-do-db= softisland

binlog-ignore-db=mysql

replicate-ignore-db=mysql

sync\_binlog=1

auto\_increment\_increment = 2

auto\_increment\_offset = 1

log-slave-updates

slave-skip-errors=all

[mysqldump]

quick

max\_allowed\_packet = 32M

[mysql]

no-auto-rehash

[myisamchk]

key\_buffer\_size = 256M

sort\_buffer\_size = 256M

read\_buffer = 2M

write\_buffer = 2M

[mysqlhotcopy]

interactive-timeout

#### 5.2.5修改/etc/my.cnf文件（节点二）

# rm -rf /etc/my.cnf

# vi /etc/my.cnf

添加如下配置代码

[client]

default-character-set = utf8

port = 3306

socket = /tmp/mysql.sock

[mysqld]

character-set-server = utf8

default-storage-engine = MyISAM

basedir = /home/softisland/mysql5.6

datadir = /home/softisland/mysqldata

log-error =/home/softisland/mysql5.6/mysql\_error.log

pid-file = /home/softisland/mysql5.6/mysql.pid

port = 3306

socket = /tmp/mysql.sock

skip-external-locking

key\_buffer\_size = 384M

max\_allowed\_packet = 32M

table\_open\_cache = 512

sort\_buffer\_size = 2M

read\_buffer\_size = 2M

read\_rnd\_buffer\_size = 8M

myisam\_sort\_buffer\_size = 64M

thread\_cache\_size = 8

query\_cache\_size = 32M

thread\_concurrency = 8

log-bin=mysql-bin

server-id = 2

binlog-do-db= softisland

replicate-do-db= softisland

binlog-ignore-db=mysql

replicate-ignore-db=mysql

sync\_binlog=1

auto\_increment\_offset = 2

auto\_increment\_increment = 2

log-slave-updates

slave-skip-errors=all

[mysqldump]

quick

max\_allowed\_packet = 32M

[mysql]

no-auto-rehash

[myisamchk]

key\_buffer\_size = 256M

sort\_buffer\_size = 256M

read\_buffer = 2M

write\_buffer = 2M

[mysqlhotcopy]

interactive-timeout

#### 5.2.6重启两台服务器的mysql（节点一和二）

# service mysqld restart

#### 5.2.7授权同步账户

给两个节点授权同步账户, 赋予REPLICATION SLAVE权限（节点一和二）

# mysql -u root -p

# mysql> GRANT REPLICATION SLAVE ON \*.\* to 'abc'@'%' identified by 'abc';

为abc用户赋予任何数据库中任何表上的REPLICATION SLAVE权限, 此用户可以在网络任意位置访问, 访问时以密码abc标记

#### 5.2.8开启监控账户

开启一个权限跟ROOT一样的远程账户用于后期脚本监控（节点一和二）

# GRANT ALL PRIVILEGES ON \*.\* TO admin@"%" IDENTIFIED BY '123456';

#### 5.2.9锁表

锁表（锁表状态下不能退出SSH，否则会失败，使用SSH可以另外克隆一个）（节点一和二）

# mysql -u root -p

# mysql> FLUSH TABLES WITH READ LOCK;

#### 5.2.10查看节点一状态

查看节点一状态（记录二进制开始文件，位置）

# mysql> show master status;

显示：

# mysql> show master status;

+------------------+----------+--------------+------------------+

| File | Position | Binlog\_Do\_DB | Binlog\_Ignore\_DB |

+------------------+----------+--------------+------------------+

| mysql-bin.000003 | 1051 | | |

+------------------+----------+--------------+------------------+

1 row in set (0.00 sec)

#### 5.2.11查看节点二状态

查看节点二状态（记录二进制开始文件，位置）

# mysql> show master status;

显示：

+------------------+----------+--------------+------------------+

| File | Position | Binlog\_Do\_DB | Binlog\_Ignore\_DB |

+------------------+----------+--------------+------------------+

| mysql-bin.000003 | 764 | | |

+------------------+----------+--------------+------------------+

1 row in set (0.00 sec)

#### 5.2.12节点一连接节点二

# mysql> CHANGE MASTER TO MASTER\_HOST='172.16.20.196',MASTER\_USER='abc', MASTER\_PASSWORD='abc',MASTER\_LOG\_FILE='mysql-bin.000003',MASTER\_LOG\_POS=764;

##### 5.2.12.1开启节点一同步进程

# mysql> START SLAVE;

##### 5.2.12.2查看节点一同步状态是否正常

# mysql> SHOW SLAVE STATUS\G;

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

……………………(省略部分)

Slave\_IO\_Running: Yes //此状态必须YES

Slave\_SQL\_Running: Yes //此状态必须YES

……………………(省略部分)

注：Slave\_IO及Slave\_SQL进程必须正常运行，即YES状态，否则都是错误的状态(如：其中一个NO均属错误)。

#### 5.2.13节点二连接节点一

# mysql> CHANGE MASTER TO MASTER\_HOST='172.16.20.195',MASTER\_USER='abc', MASTER\_PASSWORD='abc',MASTER\_LOG\_FILE='mysql-bin.000003',MASTER\_LOG\_POS=1051;

##### 5.2.13.1开启节点二同步进程

# mysql> START SLAVE;

##### 5.2.13.2查看节点二同步状态是否正常

# mysql> SHOW SLAVE STATUS\G;

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

……………………(省略部分)

Slave\_IO\_Running: Yes //此状态必须YES

Slave\_SQL\_Running: Yes //此状态必须YES

……………………(省略部分)

注：Slave\_IO及Slave\_SQL进程必须正常运行，即YES状态，否则都是错误的状态(如：其中一个NO均属错误)。

#### 5.2.14解锁节点一和节点二

# mysql> UNLOCK TABLES;

#### 5.2.15退出mysql

退出节点一和节点二的mysql，至此mysql的互为主从已经实现了

# mysql> exit;

## 六．Mysql配置文件详解

mysql配置文件详解

[client]

#设置mysql客户端默认字符集

default-character-set = utf8

#指定默认端口

port = 3306

#是个UNIX域socket文件，这种socket效率比基于TCP/IP的socket高，但是缺点是不能跨机器连接，mysql提供这个东西是为了改善本地连接的速度

socket = /tmp/mysql.sock

[mysqld]

#服务端使用的字符集

character-set-server = utf8

#创建新表时将使用的默认存储引擎

default-storage-engine = MyISAM

#基准路径，其他路径都相对于这个路径

basedir = /home/softisland/mysql5.6

# mysql数据库文件所在目录

datadir = /home/softisland/mysqldata

#错误日志存放路径

log-error = /home/softisland/mysql5.6/mysql\_error.log

#进程存放位置

pid-file = /home/softisland/mysql5.6/mysql.pid

port = 3306

socket = /tmp/mysql.sock

#不使用系统锁定，要使用myisamchk,必须关闭服务器

skip-external-locking

#myisam索引buffer,只有key没有data

key\_buffer\_size = 384M

#增长到max\_allowed\_packet字节

max\_allowed\_packet = 32M

#主要用于设置table高速缓存的数量

table\_open\_cache = 512

#排序buffer大小；线程级别

sort\_buffer\_size = 2M

#以全表扫描(Sequential Scan)方式扫描数据的buffer大小 ；线程级别

read\_buffer\_size = 2M

#MyISAM以索引扫描(Random Scan)方式扫描数据的buffer大小 ；线程级别

read\_rnd\_buffer\_size = 8M

#MyISAM 设置恢复表之时使用的缓冲区的尺寸,当在REPAIR TABLE或用CREATE INDEX创建索引或ALTER TABLE过程中排序 MyISAM索引分配的缓冲区

myisam\_sort\_buffer\_size = 64M

#线程缓存

thread\_cache\_size = 8

#查询缓存大小

query\_cache\_size = 32M

#同时运行的线程的数据 此处最好为CPU个数两倍。本机配置为CPU的个数

thread\_concurrency = 8

#以下为互为主从配置选项

#启用二进制日志

log-bin=mysql-bin

#表示是本机的序号为1,一般来讲就是master的意思。

server-id = 1

#要记录日志的数据库,如果需要多个重复设置这个选项即可

binlog-do-db=softisland

#需要复制的数据库名，如果需要多个数据库，重复设置这个选项即可

replicate-do-db= softisland

#不要记录日志的数据库名，如果需要多个重复设置这个选项即可

binlog-ignore-db=mysql

#不需要复制的数据库名，如果需要多个数据库，重复设置这个选项即可

replicate-ignore-db=mysql

#默认情况下，并不是每次写入时都将binlog与硬盘同步。因此如果操作系统或机器(不仅仅是MySQL服务器)崩溃，有可能binlog中最后的语句丢失了。要想防止这种情况，你可以使用sync\_binlog全局变量(1是最安全的值，但也是最慢的)，使binlog在每N次binlog写入后与硬盘同步。

sync\_binlog=1

#字段一次递增多少

auto\_increment\_increment = 2

#自增字段的起始值，这样A的auto\_increment字段产生的数值是：1, 3, 5, 7, …等奇数ID

auto\_increment\_offset = 1

#表示如果一个MASTER挂掉的话，另外一个马上接管；一定要加上，否则不会把更新的记录写到二进制文件里

log-slave-updates

#跳过错误，继续执行复制操作

slave-skip-errors=all

[mysqldump]

quick

max\_allowed\_packet = 32M

[mysql]

#这个配置段设置启动MySQL服务的条件；在这种情况下，no-auto-rehash确保这个服务启动得比较快。

no-auto-rehash