**mysql集群配置**

**MySQL主-从复制**

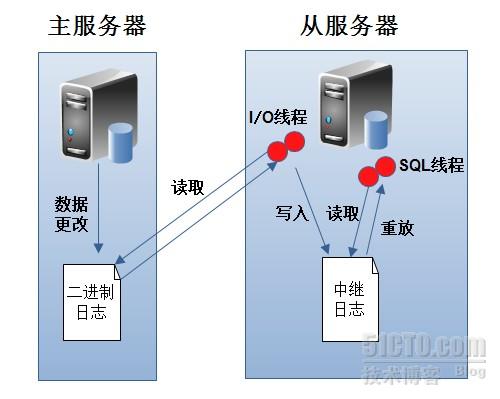
原贴：<http://rrliang.blog.51cto.com/4214966/831916>

MySQL内建的复制能力是构建大型、高性能应用程序的基础。复制解决的基本问题是让一台服务器的数据和另外的服务器保持同步。而且一台主服务器可以连接 多台从服务器，也可以使从服务器反过来作主服务器。主服务器和从服务器可以位于不同的网络拓扑中，还可以对所有数据库、特定数据库，甚至特定表进行复制。

MySQL支持的复制方案有两种：基于语句复制（Statement-Based Replication）和基于行复制（Row-Based Replication）。这两种方式的复制都是通过记录主服务器的二进制日志，并在从服务器上进行重放（Replay）完成复制的，而且都是异步的。这也意味着，从服务器的数据并非随时都是最新的。如果考虑到各种因素，从服务器的数据可能落后主服务器几秒、几分、甚至几小时都是有可能的。

需要注意的是MySQL的复制大部分都是向后兼容的。这就意味着我们构建的从服务器的MySQL版本至少要与主服务器版本相同，这样才能不出现一些意想不到的问题。

下面我们看看MySQL是如何进行复制的，如图：

[](http://rrliang.blog.51cto.com/attachment/201204/233059290.jpg)

**MySQL复制工作方式**

在主服务器上，主服务器将引起数据改变的记录都记录到二进制 日志中。然后，从服务器会启动一个工作线程，叫I/O从线程（I/O Slave Thread）。这个I/O线程开启一个普通的客户端连接到主服务器上，而主服务器会开启一个特殊的二进制转储（Binlog Dump）进程，这个转储进程从主服务器的二进制日志中读取内容，发送给从服务器。I/O线程接收发送的内容，并将其写入到中继日志中。最后，从服务器会 开启一个SQL从线程，这个线程用来读取中继日志，并且重放日志中的事件，然后更新从服务器的数据，从而完成复制。

**下面就让我们来实现一个简单的主-从复制架构吧！**

采用的环境：

两台Redhat 6.5操作系统的虚拟机；

MySQL软件：mysql-server、mysql；

假设条件：主服务器为server1，IP：172.16.14.1；从服务器server2，IP：172.16.14.2

**一、主服务器的配置**

**1、安装MySQL（从服务器的安装相同）**

为MySQL创建单独的文件系统，最好是创建逻辑卷，这对以后的磁盘容量扩展和备份很有利。这里不再给出创建过程，这里说明下我们创建的逻辑卷的名称及后续操作：

逻辑卷：/dev/myvg/mylv

挂载点：/mysql

将逻辑卷挂载至/mysql目录下，并编辑/etc/fstab文件，实现开机自动挂载；在/mysql目录中创建/mysql/data目录，作为存放数据的目录。并将其属主，属组都更改为mysql。mysql用户mysql组需要事先创建。

1. [root@server1 ~]#tar xf mysql-5.5.22-linux2.6-i686.tar.gz –C /usr/local
2. [root@server1 ~]#cd /usr/local
3. [root@server1 local]#ln –sv mysql-5.5.22-linux2.6-i686 mysql
4. [root@server1 local]#cd mysql
5. [root@server1 mysql]#cp /support-files/my-large.cnf /etc/my.cnf
6. [root@server1 mysql]#chown –R mysql:mysql .
7. [root@server1 mysql]#scripts/mysql\_install\_db --user=mysql --datadir=/mysql/data
8. [root@server1 mysql]#chown -R root .
9. [root@server1 mysql]#cp /support-files/mysql.server /etc/rc.d/init.d/mysqld
10. [root@server1 ~]#chkconfig –add mysqld
11. [root@server1 ~]#chkconfig mysqld on
12. [root@server1 ~]#service mysqld start

**2、为从服务器创建拥有相关权限的用户账号**

1. mysql**>**GRANT REPLICATION CLIENT,REPLICATION SLAVE ON \*.\* TO replay@’172.16.14.%’ IDENTIFIED BY '123456';

复制需要REPLICATION CLIENT权限；监视和管理复制需要REPLICATION SLAVE权限。

这里最好查看下二进制日志的信息：

mysql> SHOW MASTER STATUS;

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

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

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

| mysql-bin.000003 | 107 | | |

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

1 row in set (0.00 sec)

**二、从服务器的配置**

**1、编辑/etc/my.cnf配置文件**

log\_bin=mysql-bin :关闭二进制日志

server-id=10

relay\_log=relay-bin :设置中继日志的位置和文件名

因为我配置的是简单主-从架构，所以从服务器的二进制日志文件可以关闭。不过主从服务器的server-id是不能相同的。

重新启动mysql。

**2、启动复制**

mysql> CHANGE MASTER TO

MASTER\_HOST='172.16.14.1',

MASTER\_USER='replay',

MASTER\_PASSWORD='123456',

MASTER\_LOG\_FILE='mysql-bin.000003',

MASTER\_LOG\_POS=107;

mysql>START SLAVE;

mysql>SHOW SLAVE STATUS\G :查看复制的状态信息

这里实现的是主服务器没有任何数据的主-从架构，如果主服务器有大量数据该如何进行呢？稍后为您讲述！

mysql双机集群测试

原贴： <http://shugen0108.blog.51cto.com/703637/148852>

术语介绍：

管理(MGM)节点：这类节点的作用是管理MySQL簇内的其他节点，如提供配置数据、启动并停止节点、运行备份等。由于这类节点负责管理其他节点的配置，应在启动其他节点之前首先启动这类节点。MGM节点是用命令ndb\_mgmd启动的。

数据节点(ndbd)：这类节点用于保存簇的数据。数据节点的数目与副本的数目相关，是片段的倍数。例如，对于两个副本，每个副本有两个片段，那么就有4个数据节点。没有必要有一个以上的副本。数据节点是用命令ndbd启动的。

SQL节点(mysqld)：这是用来访问簇数据的节点。对于MySQL簇，客户端节点是使用NDB簇存储引擎的传统MySQL服务器。典型情况下，SQL节点是使用命令mysqld –ndbcluster启动的，或将ndbcluster添加到my.cnf后使用mysqld启动。

db1:192.168.1.188

db2:192.168.1.171

一、安装集群所需软件

db1,db2上分别执行一下命令:

shell> groupadd mysql

shell> useradd -g mysql mysql

shell> cd /usr/local

shell> gunzip < /path/to/mysql-VERSION-OS.tar.gz | tar xvf -

shell> ln -s full-path-to-mysql-VERSION-OS mysql

shell> cd mysql

shell> chown -R mysql .

shell> chgrp -R mysql .

shell> scripts/mysql\_install\_db --user=mysql

shell> chown -R root .

shell> chown -R mysql data

rpm -Uvh MySQL-Cluster-gpl-server-6.3.20-0.rhel5.i386.rpm

rpm -Uvh MySQL-Cluster-gpl-client-6.3.20-0.rhel5.i386.rpm

rpm -Uvh MySQL-Cluster-gpl-storage-6.3.20-0.rhel5.i386.rpm

rpm -Uvh MySQL-Cluster-gpl-management-6.3.20-0.rhel5.i386.rpm

rpm -Uvh MySQL-Cluster-gpl-tools-6.3.20-0.rhel5.i386.rpm

rpm -Uvh MySQL-Cluster-gpl-extra-6.3.20-0.rhel5.i386.rpm

二、配置节点

（db1,db2上分别执行）

1.config.ini文件配置

cd /var/lib/mysql-cluster

vi config.ini

在config.ini中添加如下内容：

[ndbd default]

NoOfReplicas= 2

MaxNoOfConcurrentOperations= 10000

DataMemory= 80M

IndexMemory= 24M

TimeBetweenWatchDogCheck= 30000

DataDir= /var/lib/mysql-cluster

MaxNoOfOrderedIndexes= 512

StartPartialTimeout=100

StartPartitionedTimeout=100

ArbitrationTimeout=5000

TransactionDeadlockDetectionTimeout=5000

HeartbeatIntervalDbDb=5000

StopOnError=0

[ndb\_mgmd default]

DataDir= /var/lib/mysql-cluster

[ndb\_mgmd]

Id=1

HostName= 192.168.1.188

[ndb\_mgmd]

Id=2

HostName= 192.168.1.171

[ndbd]

Id= 3

HostName= 192.168.1.188

[ndbd]

Id= 4

HostName= 192.168.1.171

[mysqld]

ArbitrationRank=2

[mysqld]

ArbitrationRank=2

[tcp default]

PortNumber= 63132

2.my.cnf文件配置

vi /etc/my.cnf

在my.cnf中添加如下内容：

[mysqld]

default-storage-engine=ndbcluster

ndbcluster

ndb-connectstring=192.168.1.188,192.168.1.171

[ndbd]

connect-string=192.168.1.188,192.168.1.171

[ndb\_mgm]

connect-string=192.168.1.188,192.168.1.171

[ndb\_mgmd]

config-file=/var/lib/mysql-cluster/config.ini

[mysql\_cluster]

ndb-connectstring= 192.168.1.188,192.168.1.171

三、启动管理节点

方法一(推荐)：

Server1为：

# ndb\_mgmd --ndb\_nodeid=1

Server2为：

# ndb\_mgmd --ndb\_nodeid=2

方法二：

Server1为：

# ndb\_mgmd -f /var/lib/mysql-cluster/config.ini

Server2为：

# ndb\_mgmd -f /var/lib/mysql-cluster/config.ini

注:不同的服务器用不同的启动命令

在启动时有一个警告提示

Cluster configuration warning:

arbitrator with id 1 and db node with id 3 on same host 192.168.1.188

arbitrator with id 2 and db node with id 4 on same host 192.168.1.171

Running arbitrator on the same host as a database node may

cause complete cluster shutdown in case of host failure.

说节点1和3，2和4的arbitrator一样，可能引起整个集群失败。（可以不用放在心上）

四、启动数据节点

方法一：初始化集群，--initial参数只有在第一次启动是使用.

在Server1中

# ndbd --nodeid=3 --initial

在Server2中

# ndbd --nodeid=4 --iniitial

方法二：

Server1为：

# ndbd --connect-string="nodeid=3;host=192.168.1.188:1186"

Server2为：

# ndbd --connect-string="nodeid=4;host=192.168.1.111:1186"

启动sql节点：

在Server1中

# mysqld\_safe --ndb\_nodeid=5 --user=mysql &

在Server2中

# mysqld\_safe --ndb\_nodeid=6 --user=mysql &

五、检查工作状态

在任意一台机子上启动管理终端：

# ndb\_mgm

键入show命令查看当前工作状态：

Cluster Configuration

---------------------

[ndbd(NDB)] 2 node(s)

id=3 @192.168.1.188 (Version: 6.0.0, Nodegroup: 0, Master)

id=4 @192.168.1.171 (Version: 6.0.0, Nodegroup: 0)

[ndb\_mgmd(MGM)] 2 node(s)

id=1 @192.168.1.188 (Version: 6.0.0)

id=2 @192.168.1.171 (Version: 6.0.0)

[mysqld(API)] 2 node(s)

id=5 (not connected, accepting connect from any host)

id=6 (not connected, accepting connect from any host)

六、检查数据是否同步

七、破坏性测试

八、集群的关闭

#ndb\_mgm -e shutdown