**MySQL-Mycat读写分离**

# 一、Mycat读写分离概述：

**Mysql作为目前世界上使用最广泛的免费数据库，相信所有从事系统运维的工程师都一定接触过。但在实际的生产环境中，由单台Mysql作为独立的数据库是完全不能满足实际需求的，无论是在安全性，高可用性以及高并发等各个方面。
因此，一般来说都是通过 主从复制（Master-Slave）的方式来同步数据，再通过读写分离（MySQL-Proxy/Amoeba）来提升数据库的并发负载能力，这样的方案来进行部署与实施的。**

## 1、Mycat读写分离原理：

**基本的原理是让主数据库处理事务的增、改、删操作（INSERT、UPDATE、DELETE），而从数据库处理SELECT查询操作。数据库复制被用来把事务性操作导致的变更同步到集群中的从数据库。**

**数据内部交换过程：**

****

**数据库集群，如果slave-1查询过多，后面进来的连接全自动切换slave-2，以此类推。**

****

## 2、为什么读写分离：

**面对越来越大的访问压力，单台的服务器的性能成为瓶颈，需要分担负载
1）、主从只负责各自的写和读，极大程度的缓解X锁和S锁争用
2）、从库可配置myisam引擎，提升查询性能以及节约系统开销**

**1、MyISAM：默认表类型，它是基于传统的ISAM类型，ISAM是Indexed Sequential Access Method (有索引的顺序访问方法) 的缩写，它是存储记录和文件的标准方法。不是事务安全的，而且不支持外键，如果执行大量的select，insert MyISAM比较适合。**

**2、InnoDB：支持事务安全的引擎，支持外键、行锁、事务是他的最大特点。如果有大量的update和insert，建议使用InnoDB，特别是针对多个并发和**[**QPS**](https://baike.baidu.com/item/QPS/4818554?fr=aladdin)**较高的情况。**

**3）、增加冗余，提高可用性**

## 3、实现读写分离的方式：

**实现读写分离一般有两种方式实现**

### 1）应用程序层实现(app代码里面写死他)

**应用程序层实现指的是在应用程序内部及连接器中实现读写分离，直接在app里面写死。没运维什么事儿了。**

****

**优点：
A：应用程序内部实现读写分离，安装既可以使用
B：减少一定部署难度
C：访问压力在一定级别以下，性能很好
缺点：
A：架构一旦调整，代码要跟着变
B：难以实现高级应用，如自动分库，分表
C：无法适用大型应用场景
D：扩容架构需要修改代码
E：无法实现自动分库分表**

### 2）中间件层实现

**[数据建库中间件层实现？](http://www.cnblogs.com/zzsdream/articles/6650690.html)**

****

**前端连接mycat。**

# 二、什么是Mycat：

**一个彻底开源的，面向企业应用开发的大数据库集群**

**支持事务、ACID、可以替代MySQL的加强版数据库**

**一个可以视为MySQL集群的企业级数据库，用来替代昂贵的Oracle集群**

**一个融合内存缓存技术、NoSQL技术、HDFS大数据的新型SQL Server**

**结合传统数据库和新型分布式数据仓库的新一代企业级数据库产品**

**一个新颖的数据库中间件产品**

# 三、服务安装与配置：

**MyCAT有提供编译好的安装包，支持windows、Linux、Mac等系统上安装与运行。官方下载主页**[**http://www.mycat.io**](http://www.mycat.io) **linux下可以下载Mycat-server-xxxxx.linux.tar.gz 解压在某个目录下，注意目录不能有空格，在Linux(Unix)下，建议放在usr/local/mycat目录下。**

## 1、目录解释如下：

**bin 程序目录，存放了window版本和linux版本，除了提供封装成服务的版本之外，也提供了nowrap的shell脚本命令，方便选择和修改，进入到bin目录：
Linux下运行：./mycat console,首先要chmod +x \*
注：mycat支持的命令{ console | start | stop | restart | status | dump }**

**conf目录下存放配置文件，server.xml是Mycat服务器参数调整和用户授权的配置文件，schema.xml是逻辑库定义和表以及分片定义的配置文件，rule.xml是分片规则的配置文件，分片规则的具体一些参数信息单独存放为文件，也在这个目录下，配置文件修改，需要重启Mycat或者通过8066端口reload.

lib目录下主要存放mycat依赖的一些jar文件。
日志存放在logs/mycat.log中，每天一个文件，日志的配置是在conf/log4j.xml中，根据自己的需要，可以调整输出级别为debug，debug级别下，会输出更多的信息，方便排查问题。**

**实验环境：部署Mycat读写分离**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **主机名** | **IP** | **系统** | **MySQL版本** | **角色** |
| **A74** | **192.168.19.74** | **CentOS7.6** | **5.7.18二进制包** | **Client** |
| **A75** | **192.168.19.75** | **CentOS7.6** | **5.7.18二进制包** | **Mycat** |
| **A76** | **192.168.19.76** | **CentOS7.6** | **5.7.18二进制包** | **Master 写** |
| **A77** | **192.168.19.77** | **CentOS7.6** | **5.7.18二进制包** | **Slave 读** |

****

## 2、配置java环境：

**安装Mycat需要安装JDK1.7版本或者更高**

**安装JDK和Mycat**

**[root@A75 ~]# rpm -qa|grep jdk**

**[root@A75 ~]# rpm -qa|grep java**

**[root@A75 ~]# mkdir /usr/java**

**[root@A75 ~]# tar fx jdk-8u181-linux-x64.tar.gz -C /usr/java/**

**配置java环境变量和mycat环境变量**

**[root@A75 ~]# vim /etc/profile**

**#set java environment**

**JAVA\_HOME=/usr/java/jdk1.8.0\_152**

**CLASSPATH=$JAVA\_HOME/lib/**

**PATH=$PATH:$JAVA\_HOME/bin**

**export PATH JAVA\_HOME CLASSPATH**

**使环境变量生效并查看**

**[root@A75 ~]# source /etc/profile**

**[root@A75 ~]# java -version**

**java version "1.8.0\_181"**

**Java(TM) SE Runtime Environment (build 1.8.0\_181-b13)**

**Java HotSpot(TM) 64-Bit Server VM (build 25.181-b13, mixed mode)**

**[root@A75 ~]#**

## 3、安装配置Mycat：

**[root@A75 ~]# tar fx Mycat-server-1.5.1-RELEASE-20161130213509-linux.tar.gz -C /usr/local/**

**[root@A75 ~]# vim /etc/profile**

**#MyCAT在Linux中部署启动时，首先需要在Linux系统的环境变量中配置MYCAT\_HOME。加入到环境变量。**

**MYCAT\_HOME=/usr/local/mycat PATH=$MYCAT\_HOME/bin:$PATH**

**[root@A75 ~]# source /etc/profile**

**[root@A75 ~]# mycat**

**Usage: /usr/local/mycat/bin/mycat { console | start | stop | restart | status | dump }**

**如果是在多台Linux系统中组建的MyCAT集群，那需要在MyCAT Server所在的服务器上配置对其他ip和主机名的映射，配置方式如下：**

**vim /etc/hosts**

**192.168.19.75 A75**

**配置mycat(mycat的用户账号和授权信息是在conf/server.xml文件中配置**

**[root@A75 ~]# vim /usr/local/mycat/conf/server.xml**

**<!--配置文件里面的账号密码是给前端的，告诉前端能用那些账号连接数据库。也就是图上的mysql-cilent。-->**

**<!--此处需要注意的是：是在34行到43行修改文件。(千万不要全部删除替换)-->**

**</system>**

**<user name="root"> <!--这个账号就是用来给前端登录mycat的账号的。定义一下->**

 **<property name="password">123456</property>**

 **<property name="schemas">test</property> <!--定义操作的那个数据库->**

**</user>**

**<user name="user">**

 **<property name="password">user</property>**

 **<property name="schemas">test</property>**

 **<property name="readOnly">true</property> <!--只读->**

**</user>**

**编辑MyCAT的配置文件schema.xml，关于dataHost的配置信息如下：**

**[root@A75 ~]# cp /usr/local/mycat/conf/schema.xml /usr/local/mycat/conf/schema.xml.bak**

**[root@A75 ~]# >/usr/local/mycat/conf/schema.xml**

**[root@A75 ~]# vim /usr/local/mycat/conf/schema.xml**

**<!--配置文件里面的账号密码是什么意思？-->**

**<!--是mycat这台服务器去操作数据库用到的账号，这个账号必须在数据库创建好。-->**

**<?xml version="1.0"?>**

**<!DOCTYPE mycat:schema SYSTEM "schema.dtd">**

**<mycat:schema xmlns:mycat="http://org.opencloudb/">**

 **<schema name="test" checkSQLschema="false" sqlMaxLimit="100" dataNode='dn1'>**

 **</schema>**

 **<dataNode name="dn1" dataHost="dthost" database="test"/>**

 **<dataHost name="dthost" maxCon="500" minCon="10" balance="1" writeType="0" dbType="mysql" dbDriver="native" switchType="-1" slaveThreshold="100">**

 **<heartbeat>select user()</heartbeat>**

 **<writeHost host="A76" url="192.168.19.76:3306" user="mycat" password="123456">**

 **</writeHost>**

 **<writeHost host="A77" url="192.168.19.77:3306" user="mycat" password="123456" />**

 **</dataHost>**

**</mycat:schema>**

**保存退出，这就配置完了。但是不要启动，因为还没有配置完呢，上面schema.xml的配置文件mycat账号还没有创建呢。**

**有两个参数需要注意，balance和 switchType。
其中，balance指的负载均衡类型，目前的取值有4种：**

|  |  |
| --- | --- |
| **1、balance="0"** | **不开启读写分离机制，所有读操作都发送到当前可用的writeHost上** |
| **2、balance="1"** | **全部的readHost与stand by writeHost参与select语句的负载均衡，简单的说，当双主双从模式(M1->S1，M2->S2，并且M1与 M2互为主备)，正常情况下，M2,S1,S2都参与select语句的负载均衡** |
| **3、balance="2"** | **所有读操作都随机的在writeHost、readhost上分发。** |
| **4、balance="3"** | **所有读请求随机的分发到wiriterHost对应的readhost执行，writerHost不负担读压力** |

**switchType指的是切换的模式，目前的取值也有4种**

|  |  |
| --- | --- |
| **1、switchType='-1'** | **表示不自动切换** |
| **2、switchType='1'** | **默认值，表示自动切换** |
| **3、switchType='2'** | **基于MySQL主从同步的状态决定是否切换,心跳语句为 show slave status** |
| **4、switchType='3'** | **基于MySQL galary cluster的切换机制（适合集群）（1.4.1），心跳语句为 show status like 'wsrep%'** |

## 4、配置MySQL主从：

**配置A76、A77MySQL主从。**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **主机名** | **IP** | **系统** | **MySQL版本** | **角色** |
| **A76** | **192.168.19.76** | **CentOS7.6** | **5.7.18二进制包** | **Master** |
| **A77** | **192.168.19.77** | **CentOS7.6** | **5.7.18二进制包** | **Slave** |

**创建需要同步的库，并且主从数据库同时插入一些数据(用于同步使用便于区分)。**

**mysql> create database test;**

**mysql> use test;**

**mysql> create table emp (empno numeric(4) not null,ename varchar(10),job varchar(9),mgr numeric(4),hiredate datetime,sal numeric(7, 2),comm numeric(7, 2),deptno numeric(2));**

**mysql> insert into emp values (7369, 'SMITH', 'CLERK', 7902, '1980-12-17', 800, null, 20);**

**mysql> select \* from emp;**

****

**修改A75配置文件。**

**[root@A76 ~]# vim /etc/my.cnf**

**#master and slave**

**log-bin=mysql-bin-master**

**server-id=1**

**binlog-do-db=test**

**binlog-ignore-db=mysql**

**[root@A76 ~]# /etc/init.d/mysqld restart**

**在A76上授权一个账号只有复制权限让A77同步使用并查看show master状态信息**

**[root@A76 ~]# mysql -uroot -p123456**

**mysql> grant replication slave on \*.\* to slave@192.168.19.77 identified by "123456";**

**mysql> select \* from mysql.user where user='slave'\G;**

**mysql> flush privileges;**

**mysql> show master status;**

****

**需要记住上面圈主红色框框的信息。一会儿会在A77配置从库change master的时候会使用到fire的名字及position号。**

**修改A77mysql配置文件并重启并到A77从库测试能否登录A76mysql**

**[root@A77 ~]# vim /etc/my.cnf**

**#master and slave**

**server-id=2**

**[root@A77 ~]# /etc/init.d/mysqld restart**

**[root@A77 ~]# mysql -uslave -p123456 -h192.168.19.76 -e "show databases;"**

****

**#上面的回显结果只能看到一个库。这是对的。因为这个账号只有同步的权限。**

**A77从库配置主从，也就是chang master 以确定需要同步的ip、用户名、密码、binlog文件、position号**

**[root@A77 ~]# mysql -uroot -p123456**

**mysql> stop slave;**

**mysql> change master to master\_host='192.168.19.76',master\_port=3306,master\_user='slave',master\_password='123456',master\_log\_file='mysql-bin-master.000001',master\_log\_pos=602;**

**mysql> start slave;**

**mysql> show slave status \G 查看状态**

 **Slave\_IO\_Running: Yes**

 **Slave\_SQL\_Running: Yes**

**以上两个线程都为yes主从成功搭建完毕。一个sql线程一个io线程。**

**A76主库插入数据测试同步。A77从库查询是否自动同步。**

**[root@A76 ~]# mysql -uroot -p123456**

**mysql> insert into test.emp values (7499, 'ALLEN', 'SALESMAN', 7698, '1981-02-20', 1600, 300, 30);**

**mysql> select \* from test.emp;**

****

## 5、创建Mycat登录后端数据库账号：

**在主从两台服务器创建mycat账号。用于mycat服务器登录数据库使用。否则无法建立读写分离。必须要创建。**

**[root@A76、A77 ~]# mysql -uroot -p123456**

**mysql> grant all on \*.\* to mycat@'192.168.19.75' identified by '123456';**

**mysql> flush privileges;**

## 6、启动Mycat并查看端口：

**启动Mycat并查看日志**

**[root@A75 ~]# mycat start**

**Starting Mycat-server...**

**[root@A75 ~]# cat /usr/local/mycat/logs/wrapper.log**

****

**查看端口**

**[root@A75 ~]# netstat -antlp|grep 8066**

**tcp6 0 0 :::8066 :::\* LISTEN 18249/java**

## 7、测试Mycat读写分离：

**在mysql-client测试(登录mycat)  #注意第四行回显。为mycat-1.5.1。**

**[root@A74 ~]# mysql -uroot -p123456 -h192.168.19.75 -P8066**

****

****

**在mycat服务器修改日志输出模式。(测试读写是否分离)**

**[root@A75 ~]# vim /usr/local/mycat/conf/log4j.xml**

**37行info修改成debug模式**

**<level value="info" /> <level value="debug" />**

**debug是调试模式**

**客户端进行写入或者查询数据
实时监测mycat日志进行测试读写分离**

**测试查询是否为A77 mysql-slave服务器**

**[root@A75 ~]# tail -f /usr/local/mycat/logs/mycat.log**

****

**测试查询是否为A76 mysql-master服务器**

****