**MyCAT使用经验分享**

 通过对mycat一个月左右的研究和使用，目前已经正式应用于我公司正式项目中，为了感谢群里各位朋友的帮助，现对mycat做一下使用经验分享。

## 环境准备

 mycat是使用java语言开发的，需要先安装java运行环境，由于mycat中使用了JDK7中的一些特性，所以要求JDK7以上的版本才能运行。

**JDK下载：**

<http://www.java.com/zh_CN/>

**MyCAT最新开源版本下载，项目主页：**

<https://github.com/MyCATApache/>

**MyCAT 二进制包下载：**

<https://github.com/MyCATApache/Mycat-download>

**MyCAT 文档下载：**

<https://github.com/MyCATApache/Mycat-doc>

**MySQL 下载：**

<http://dev.mysql.com/downloads/mysql/5.5.html#downloads>

(注：MyCAT支持多种数据库接入，如：MySQL、SQL Server、Oracle、MongoDB等. 推荐使用MySQL做集群.)

## 环境安装与配置

 如果是第一次刚接触MyCat建议下载源码在本地通过eclipse等工具进行配置和运行，便于深入了解和调试程序运行逻辑。

**1）源代码方式调试与配置**

 由于MyCat源代码目前主要托管在github上，大家需要在本地安装和配置好相关环境，具体参考群共享中 “github-eclipse开发指南.docx”说的很详细，按照文档中的步骤把源代码下载到本地即可。

**MyCAT目前通过配置文件的方式来定义逻辑库和相关配置：**

* MYCAT\_HOME/conf/schema.xml中定义逻辑库，表、分片节点等内容
* MYCAT\_HOME/conf/rule.xml中定义分片规则
* MYCAT\_HOME/conf/server.xml中定义用户以及系统相关变量，如端口等。

**运行：**

 入口程序是org.opencloudb.MycatStartup.java，右键run as 出现下面的界面，需要设置MYCAT\_HOME目录，为你工程当前所在目录(src/main):



设置完MYCAT主目录后即可正常运行MyCAT服务。

注：若启动报错，DirectBuffer内存不够，则可以再加JVM系统参数：

XX:MaxDirectMemorySize=128M

**2) 使用MyCAT 服务安装包方式运行与配置**

 MyCAT 有提供编译好的安装包，支持windows、Linux、Mac、Solaris等系统上安装与运行。

windows下可以下载Mycat-server-xxxxxxx-win.tar.gz

linux下可以下载Mycat-server-xxxxxxx.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，如下:

![OW)T8@W~VX(S]ZI26L594CJ]()

下面是修改MyCAT用户密码的方式(仅供参考)：





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，方便大家选择和修改，进入到bin目录：

* Windows 下 运行: mycat.bat console 在控制台启动程序，也可以装载成服务，若此程序运行有问题，也可以运行startup\_nowrap.bat，确保java命令可以在命令执行。
* Linux下运行：mycat console,首先要chmod +x \*

Warp方式的命令，可以安装成服务并启动或停止**(做成服务的方式)**。

* mycat install (可选)
* mycat start

**注意，wrap方式的程序，其JVM配置参数在conf/wrap.conf中，可以修改为合适的参数，参数调整参照**[**http://wrapper.tanukisoftware.com/doc/english/properties.html**](http://wrapper.tanukisoftware.com/doc/english/properties.html)

 **c**onf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9066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。

## 数据库映射与配置

**MyCAT逻辑库相关概念：**

* **schema**:逻辑库，与MySQL中的Database（数据库）对应，一个逻辑库中定义了所包括的Table。
* **table**：表，即物理数据库中存储的某一张表，与传统数据库不同，这里的表格需要声明其所存储的逻辑数据节点DataNode，这是通过表格的分片规则定义来实现的，table可以定义其所属的“子表(childTable)”，子表的分片依赖于与“父表”的具体分片地址，简单的说，就是属于父表里某一条记录A的子表的所有记录都与A存储在同一个分片上。
* **分片规则**：是一个字段与函数的捆绑定义，根据这个字段的取值来返回所在存储的分片（DataNode）的序号，每个表格可以定义一个分片规则，分片规则可以灵活扩展，默认提供了基于数字的分片规则，字符串的分片规则等。
* **DataNode**：**MyCAT**的逻辑数据节点，是存放table的具体物理节点，也称之为分片节点，通过DataSource来关联到后端某个具体数据库上，一般来说，为了高可用性，每个DataNode都设置两个DataSource，一主一丛，当主节点宕机，系统自动切换到从节点。
* **DataSource**：定义某个物理库的访问地址，用于捆绑到Datanode上。

编辑MYCAT\_HOME/conf/schema.xml文件，增加对物理数据表映射信息：



**table节点中各属性说明：**

 name：指定物理表名称；

 primaryKey：指定物理表主键名称；

 autoIncrement：指定主键是否采用自增长策略，如果不需要自增长可不用配置；

 dataNode：指定表所在的逻辑数据节点，如果有多个节点可使用逗号分隔，也可使用表 达式dn$0-99 指定在0-99这100个节点上都使用(如果是一个节点上有多个库可使 用db1$0-99)；

 rule：指定路由规则；

**逻辑数据节点与物理主机配置：**



**dataNode节点中各属性说明：**

 name：指定逻辑数据节点名称；

 dataHost：指定逻辑数据节点物理主机节点名称；

 database：指定物理主机节点上。如果一个节点上有多个库，可使用表达式db$0-99， 表示指定0-99这100个数据库；

dataHost 节点中各属性说明：

 name：物理主机节点名称；

 maxCon：指定物理主机服务最大支持1000个连接；

 minCon：指定物理主机服务最小保持10个连接；

 writeType：指定写入类型；

* 0，只在writeHost节点写入；
* 1，在所有节点都写入。慎重开启，多节点写入顺序为默认写入根据配置顺序，第一个挂掉切换另一个；

 dbType：指定数据库类型；

 dbDriver：指定数据库驱动；

 balance：指定物理主机服务的负载模式。

* 0，不开启读写分离机制；
* 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；
* 2，所有的readHost与writeHost都参与select语句的负载均衡，也就是说，当系统的写操作压力不大的情况下，所有主机都可以承担负载均衡；

一个dataHost元素，表明进行了数据同步的一组数据库，**DBA需要保证这一组数据库服务器是进行了数据同步复制的**。writeHost相当于Master DB Server，而其下的readHost则是与从数据库同步的Slave DB Server。当dataHost配置了多个writeHost的时候，任何一个writeHost宕机，Mycat 都会自动检测出来，并尝试切换到下一个可用的writeHost；

**MyCAT支持高可用性的企业级特性，根据您的应用特性，可以配置如下几种策略：**

* 后端数据库配置为一主多从，并开启读写分离机制；
* 后端数据库配置为双主双从（多从），并开启读写分离机制；
* 后端数据库配置为多主多从，并开启读写分离机制；

后面两种配置，具有更高的系统可用性，当其中一个写节点（主节点）失败后，Mycat会侦测出来（心跳机制）并自动切换到下一个写节点，**MyCAT在任何时候，只会往一个写节点写数据;**

**下面是典型的双主双从的Mysql集群配置：**



## sequence设置与使用

 在实现分库分表的情况下，MySql自增主键已无法保证自增主键的全局唯一，为此，MyCat 提供了全局sequence，并使用两种机制实现这种全局sequence，分别是文件形式和数据库形式。

**1）sequence使用方式**

 Sequence名称：TEST\_SEQ；

 使用方式：select next value for MYCATSEQ\_TEST\_SEQ；

 注意，使用时sequence名称前一定要加大写的“MYCATSEQ\_”；

**2）文件形式**

 原理：此种方式MyCat将sequence配置到文件中，当使用到配置中的sequence后，MyCat会更新classpath中的sequence\_conf.properties文件中的sequence当前值；

**配置方式：**

 在sequence\_conf.properties文件中做如下配置：

 # self define sequence

TEST\_SEQ.HISIDS=

TEST\_SEQ.MINID=1001

TEST\_SEQ.MAXID=2000

TEST\_SEQ.CURID=1000

 其中MINID：最小值； MAXID：最大值；CURID：当前值；

 server.xml配置

<system>

<property name="sequnceHandlerType">0</property>

</system>

 sequnceHandlerType配置为0，表示使用文件形式；

**缺点：**

 当MyCat重新发布后，sequence会恢复到初始值；

**3）数据库形式**

 原理：此种方式是在数据库中建立一张表，存放sequence名称(name)，sequence当前值(current\_value)，步长（increment int类型每次读取多少个sequence，假设为K）等信息；

**Sequence获取步骤：**

❶、当初次使用该sequence时，根据传入的sequence名称，从数据库这张表中读取 current\_value，和increment到MyCat中，并将数据库中的current\_value设置为原 current\_value值+increment值；

❷、MyCat将读取到current\_value+increment作为本次要使用的sequence值，下次使 用时，自动加1，当使用increment次后，执行步骤❶相同的操作；

MyCat负责维护这张表，用到哪些sequence，只需要在这张表中插入一条记录即可。若某次读取的sequence没有用完，系统就停掉了，则这次读取的sequence剩余值不会在使用；

**配置方式：**

 **server.xml配置**

<system>

<property name="sequnceHandlerType">1</property>

</system>

 sequnceHandlerType配置为1，表示使用数据库形式；

 **数据库配置:**

 建立sequence相关表和函数，具体如附件:

 

 **sequence\_db\_conf.properties文件配置：**

 指定sequence相关表和函数在哪个节点上：

 Test\_SEQ=dn1

**4）sequence使用**

目前mycat中支持配置全局sequence和自定义sequence两种方式。

 **使用默认的全局sequence :**

 insert into tb1(id,name) values(next value for MYCATSEQ\_GLOBAL,'micmiu.com');

 **使用自定义的 sequence :**

insert into tb2(id,name) values(next value for MYCATSEQ\_MY1,'micmiu.com');

 注：在没有特殊需求的情况下，建议每个表都使用自定义的sequence，每个表配置一个，这样比较方便管理和维护；

**5）autoIncrement开启**

 在mycat中使用主键自增长需要在schema.xml的table节点中的autoIncrement设置为true表示开启主键自增长策略；

## catlet支持

 通过catlet支持跨分片复杂SQL实现以及存储过程支持等。使用方式主要通过mycat注释的方式来执行，如下：

**1. 跨分片联合查询注解支持：**

 /\*!mycat:catlet=demo.catlets.ShareJoin \*/ select bu.\*,sg.\* from base\_user bu,sam\_glucose sg where bu.id\_=sg.user\_id；

注：sam\_glucose 是跨分片表；

**2. 存储过程注解支持：**

 /\*!mycat: sql=select \* from base\_user where id\_=1;\*/CALL proc\_test();

 注：目前执行存储过程通过mycat注解的方式执行，注意需要把存储过程中的sql写到注解中；

## MySQL主从配置

 在使用mycat管理一个mysql集群的时候经常需要使用主从数据复制的功能，以下是基于mysql5.5以上版本最新的主从配置：

开启mysql主从数据复制，主要在mysql的my.ini文件中设置：

windows系统一般在c:\ProgrameData\MySQL中；

Linux系统一般在 /etc/my.cnf中；

**1、主数据库服务器配置：**

server-id=1

log-bin=mysql-bin

binlog-do-db=sam\_test       //只同步sam\_test数据库

binlog-ignore-db=mysql    //避免同步mysql用户配置，以免不必要的麻烦

**2、创建一个允许从服务器来访问的用户(主服务器)：**

grant replication slave on \*.\* to 'p1test'@'192.168.18.74' identified by '123456';

**3、从数据库服务器配置：**

server-id=101

log-bin=mysql-bin

replicate-do-db=sam\_test     //指定只同步的数据库

replicate-ignore-db=mysql   //屏蔽对mysql用户配置的同步，以免不必要的麻烦

**4、在从数据库服务器上给主服务器授权：**

注意：mysql5.1版本以上，从服务器上my.ini中不支持配置master-host= ..... 此类配置；

需要采用：

mysql> stop slave;

mysql> change master to master\_host='192.168.18.78',master\_user='p1test',master\_password='123456';

mysql> slave start;

**查看主服务数据同步状态：**

show master status;

或 show master status\G;

**查看从服务数据同步状态：**

show slave status;

或 show slave status\G;

**在学习和使用mycat过程中主要参考的文档有：**

MyCat\_In\_Action\_中文版.doc

Mycat\_Develope\_Guide.docx

Mycat最权威Mysql主从配置手册.docx

mycat路由解析开发指南.docx

MyCat Sequence机制.docx

MyCat Sequence配置.sql

github-eclipse开发指南.docx

github个人使用笔记.doc

MyCat不支持的Sql语句.docx

MyCat生产部署经验分享.docx

分布式数据库选型.pptx

注：以上文档群共享中都有，如果大家对mycat的感兴趣，希望大家都能认真的看看这些文档，感谢群里这么多朋友的分享，也希望在未来有更多的人参与到mycat社区中！

 兵临城下 2015-02-08

 QQ：465456975