**Mycat指南**

**入门篇**

1. **安装**

1.1从<https://github.com/MyCATApache/Mycat-download>下载压缩包

1.2解压缩后复制到相应目录下面，比如/usr/local/mycat

1.3mycat目录下面分别是bin conf lib logs。

bin目录里是启动脚本

conf目录里是配置文件

lib目录里是Mycat和它的依赖jar

logs目录里是console.log用来保存控制台日志，和mycat.log用来保存mycat的log4j日志

1. **配置环境变量**

指定指定MYCAT\_HOME，比如按照前文所述，可以把MYCAT\_HOME配置为：export MYCAT\_HOME=/usr/local/mycat

1. **启动**

Mycat有两种启动方式，第一种是运行$MYCAT\_HOME/bin/mycat脚本执行wrap模式的启动；第二种是运行$MYCAT\_HOME/bin/startup\_nowrap.sh执行nowrap模式的启动

1. **Mycat类加载目录与JVM参数**

以nowrap模式为例，mycat会加载$MYCAT\_HOME/lib/classes和$MYCAT\_HOME/lib目录下的所有jar。

如果要修改jvm启动参数，可以修改$MYCAT\_HOME/bin/startup\_mycat.sh里的JAVA\_OPTS变量

1. **配置文件**

重点介绍rule.xml schema.xml router.xml server.xml log4j.xml

* 1. rule.xml

分片规则配置文件，mycat支持的所有分片规则都在这个文件里。定义一个规则需要两个节点，一个是tableRule，一个是function。

以一致性哈希分片为例，function节点定义了分片规则的实现类与初始化参数和分片规则的算法名称。

属性name为规则算法名称

class为规则实现类

property子节点为初始化规则的参数，使用seed count virtualBucketTimes就可初始化一致性哈希规则

<function name="murmur" class="org.opencloudb.route.function.PartitionByMurmurHash">

<property name="seed">0</property><!-- 默认是0-->

<property name="count">2</property><!-- 要分片的数据库节点数量，必须指定，否则没法分片-->

<property name="virtualBucketTimes">160</property><!-- 一个实际的数据库节点被映射为这么多虚拟节点，默认是160倍，也就是虚拟节点数是物理节点数的160倍-->

</function>

tableRule节点定义了分片规则名（**注意此处是规则名，前面的function节点的name属性是算法名**）rule子节点指定用来分片的数据库表字段和**分片算法名**，也就是前面的function节点的name属性。

<tableRule name="sharding-by-murmur">

<rule>

<columns>id</columns>

<algorithm>murmur</algorithm>

</rule>

</tableRule>

任何自定义的分片规则也可以这样配置

* 1. server.xml

此文件用来配置mycat全局参数

**<system>节点**

<property name="defaultSqlParser">druidparser</property>，指定SQL解析器，默认是fdbparser，经测试druidparser效率更高

<property name="sequnceHandlerType">1</property>，指定mycat自动序列号生成方式。0：在借助本地文件生成序列号，1：借助数据库生成序列号，更多信息请参考Mycat in action

<property name="serverPort">8066</property>指定mycat服务端口号，mycat通过这个端口接收数据库客户端的访问请求。

另外还有一个9066端口没有出现在配置文件中，用来接收mycat监控命令、查询mycat运行状况、重新加载配置文件等。更多信息请参考Mycat in action。

**<user>节点**

name属性指定mycat用户名

<property name=”password”>password</property><!—用户密码-->

<property name=”schema”>database\_name</property><!—数据库名，用客户端建立连接后要操作的数据库名 -->

<property name=”readOnly”>false</property><!—指定是不是只读库-->

可以有多个user节点。

* 1. router.xml

<queryRouter schema=”schema”><!—没太搞清楚这个文件的意义，我把schema属性、<name>、<queryNode>都配置成了一样的，而且都与要连接的数据库名保持一致 -->

<!—schema的值会在schema.xml用到 -->

<dataNode>

<name>dataNodeName</name>

<queryNode>queryNode</queryNode>

</dataNode>

</queryRouter>

* 1. schema.xml

<schema name=”schema”><!—就是router.xml命名的schema-->

<table name=”table\_name” <!—要与它代理的数据库物理表名一致-->

primaryKey=”pk” <!—被用作主键的字段名-->

dataNode=”dataNodeName” <!—下面将会出现的dataNode节点-->

rule=”sharding-by-murmur”<!—rule.xml的tableRule节点name属性值，指定这个节点表示用这个规则执行分片-->

authIncrement=”true”><!—插入数据时是否由mycat自动生成分片，指定true时，如果insert sql内没有指定主键的值，而主键又是自增长的，mycat会自动插入生成主键的代码，并在生成主键时按照指定的分片规则将数据保存到数据库-->

<!—mycat有自己的自增序列命令，更多信息请参考Mycat in action-->

<!—此处可以指定多个table子节点，一个table节点表示一张表-->

</schema>

<dataNode name=”dataNodeName” dataHost=”localhost“ database=”database”/>

<!—分别是节点名，就是schema节点里的table子节点dataNode属性

dataHost是节点主机名，在下面要出现的dataHost定义

database数据库名

-->

<dataHost name=”localhost” maxCon=”1000” minCon=”10” balance=”0”

wrteType=”0” dbType=”mysql” dbDriver=”native>

<!—name属性是dataHost名

maxCon minCon分别是连接到物理数据库的最大最小连接数

dbType指定数据库类型

dbDriver只有两个取值，分别是native和jdbc。native为mycat自带驱动，只支持mysql，jdbc为使用jdbc实现连接数据库，指定什么数据库的jdbc驱动就可以访问什么数据库，更灵活，但效率不如native

可以有多个dataHost节点

-->

<heartbeat>select 1</heartbeat><!—保持连接不断的心跳sql-->

<writeHost host=”hostM1” url=”localhost:3306” user=”user” password=”password”>

<reeadHost host=”hostS1” url="localhost:3306" user="root" password="123456"/>

</writeHost>

<!—可以有多个writeHost，一个writeHost里要吧有多个readHost-->

</dataHost>

**现在所有配置已经完成可以执行bin目录下的启动脚本完成启动了。然后就可以像使用真正的数据库一样访问mycat。如果运行时修改了配置文件，可以用在命令行用mysql命令登录9066端口，执行show @@help得到相关命令，其中有重新加载全部配置文件的命令，可以运行时重新加载配置**

1. **二次开发**
   1. 开发新的分片规则

建一个新的类，继承AbstractPartitionAlgorithm实现RuleAlgorithm。重写public void init()和public Integer calculate(String columnValue)。

init根据rule.xml指定的分片初始化参数初始化分片规则，calculate(String)接收分片字段的字符串形式计算记录应该保存的节点。

1. **升级注意事项**
   1. 从1.2.\*到1.3.\*的变化

7.1.1 server.xml 的<property name="processors">32</property>，通常设置为CPU核数的2倍，比如CPU核数是16，此处就设置为32；<property name="processorExecutor">32</property>，默认不用设置了，在1.3.\*已经没有重要影响

7.1.2 对分片规则的类名改了一些拼写错误，注意升级rule.xml。也可以自己修改，变化在地原来的Partion改为Partition。