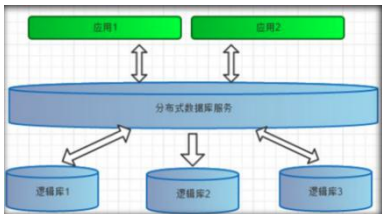


## 10\_RDMBS203 数据分片概述部署 mycat 服务测试配置

### 一 数据分片概述

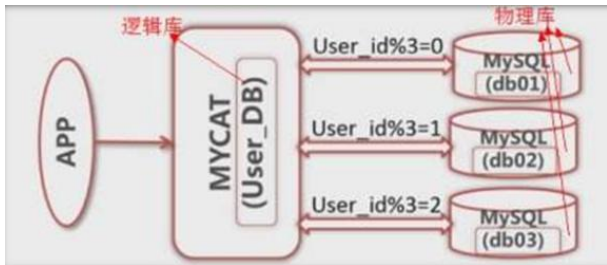
#### 1.1 分库/分表



定义:将存放在一台数据服务器中的数据,按照特定方式进行拆分,分散存放到多台数据库服务器中,以达到分散单台服务器负载的效果

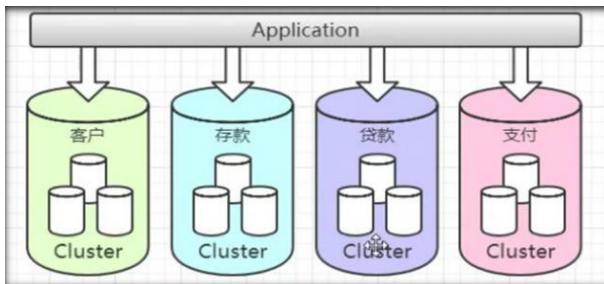
#### 1.2 分隔方式-水平分隔

横向切分,按照表中指定字段的分片规则,将表记录按行切分,分散存储到多个数据库中。



### 1.3 分隔方式-垂直分隔

纵向切分,将单个数据库的多个表按业务类型分类,分散存储到不同的数据库。



## 二 mycat 介绍

### 2.1 介绍

mycat 是基于 java 的分布式数据库系统中间件,为高并发环境的分布是存储提供解决方案。

适合数据大量写入的存储要求

支持 mysql oracle sqlserver mongodb

提供数据读写分离服务

提供数据分片服务

基于阿里巴巴 cobar 进行研发的开源软件

### 2.2 mycat 支持提供 10 种分片规则

枚举法

sharding-by-intfile

固定分片	rule1
范围约定	auto-sharding-long
求模法	mod-long
日期列分区法	sharding-by-date
通配取模	sharding-by-pattern
ASCII 码求模通配	sharding-by-prefixpattern
编程指定	sharding-by-substring
字符串拆分 hash 解析	sharding-by-stringhash
一致性 hash	sharding-by-murmur

## 2.3 工作流程

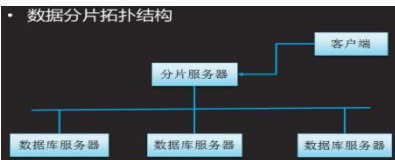
当 mycat 收到一个 SQL 命令时

2.3.1 解析 SQL 命令涉及到的表

2.3.2 然后看对表的配置,如果有分片规则,则获取 SQL 命令里分片字段的值,并匹配分片函数,获得分片表

2.3.3 然后将 SQL 命令发往对应的分片服务器去执行

2.3.4 最后收集和处理所有分片结果数据,并返回给客户端。



主机名	角色	数据库	IP地址
client50	客户端	无	192.168.4.50/24
mycat56	分片服务器	无	192.168.4.56/24
mysql53	数据库服务器	db1	192.168.4.53/24
mysql54	数据库服务器	db2	192.168.4.54/24
mysql55	数据库服务器	db3	192.168.4.55/24

### 三 部署 mycat 服务(56 上安装配置 mycat)

#### 3.1 安装 jdk-系统自带的 java-1.8.0-openjdk

```
mysql56 ~]# yum -y install java-1.8.0-openjdk #安装依赖包
```

```
mysql56 ~]# which java #查看 Java 安装路径
```

```
mysql56 ~]# java -version #查看 Java 版本
```

```
room9pc01 mysql]$ scp Mycat-server-1.6-RELEASE-20161028204710
-linux.tar.gz root@192.168.4.56:/root #传输 mycat 安装包
```

#### 3.2 安装 mycat 服务软件包

```
mysql56 ~]# tar -zxvf Mycat-server-1.6-RELEASE-20161028204710
-linux.tar.gz
```

```
mysql56 ~]# mv mycat /usr/local/
```

```
mysql56 ~]# ls /usr/local/mycat
```

```
bin  catlet  conf  lib  logs  version.txt
```

#### 3.3 mycat 目录结构/usr/local/mycat 下

bin	目录	存放 mycat 命令文档
catlet	目录	存放扩展功能文档
conf	目录	存放配置文档
lib	目录	存放 mycat 使用的 jar 包(jar 包为写好的 java 程序打包)
logs	目录	存放 mycat 启动日志和运行日志

logs/wrapper.log            mycat 服务启动日志(程序启动后产生)

logs/mycat.log            记录 SQL 脚本执行后的报错内容(程序启动后产生)

### 3.4 重要配置文件说明(conf 目录内以.xml 结尾的 mycat 服务配置文件)

server.xml            设置连接账号及逻辑库

schema.xml            配置数据分片

rule.xml            分片规则

其他文件            函数调用文件(.txt 和.properties 结尾的是分片规则文件)

### 3.5 修改配置文件

#### 3.5.1 配置客户端连接数据时使用的用户名和密码

mysql56 ~]# vim /usr/local/mycat/conf/server.xml

使用默认配置,不作修改

```
<user name="root">        #连接 mycat 服务用户名

                         <property name="password">123456</property>        #用户密码

                         <property name="schemas">TESTDB</property>        #逻辑库名

</user>
```

```

<user name="user">

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

    <property name="schemas">TESTDB</property>

    <property name="readOnly">true</property>    #只读权限

</user>

```

### 3.5.2 定义数据分片

**<schema>.....</schema>**      #定义分片信息,定义库名

**<table>.....</table>**      #定义表

**name**      #逻辑库名或逻辑表名

**dataNode**      #指定数据节点名

**rule**      #指定使用的分片规则

**type=global**      #数据不分片存储

mysql56 conf]# **cp schema.xml{,.bak}**      #备份

mysql56 conf]# **sed -i '56,77d' schema.xml**      #删除多余行

mysql56 conf]# **sed -i '39,42d' schema.xml**

mysql56 conf]# **sed -i '16,18d' schema.xml**

mysql56 conf]# **sed -i '45,47d' schema.xml**

mysql56 conf]# **wc -l schema.xml**

45 schema.xml      #删除多余行,剩 45 行

mysql56 conf]# **vim schema.xml**      #打开,显示行号,进行修改

### 3.5.3 定义分片的表,配置<schema>.....</schema>段

```
<schema name="TESTDB" checkSQLschema="false" sqlMaxLimit="100">

  <table name="travelrecord" dataNode="dn1,dn2,dn3" rule="auto-sharding-long" />

  <table name="company" primaryKey="ID" type="global" dataNode="dn1,dn2,dn3" />

  <table name="goods" primaryKey="ID" type="global" dataNode="dn1,dn2,dn3" />

  <table name="hotnews" primaryKey="ID" autoIncrement="true"

dataNode="dn1,dn2,dn3"

        rule="mod-long" />

  <table name="employee" primaryKey="ID" dataNode="dn1,dn2,dn3"

        rule="sharding-by-intfile" />

  <table name="customer" primaryKey="ID" dataNode="dn1,dn2,dn3"

        rule="sharding-by-intfile">

    <childTable name="orders" primaryKey="ID" joinKey="customer_id"

          parentKey="id">

      <childTable name="order_items" joinKey="order_id"

            parentKey="id" />

    </childTable>

    <childTable name="customer_addr" primaryKey="ID" joinKey="customer_id"

          parentKey="id" />

  </table>
```

</schema>

### 3.5.4 定义数据节点

<dataNode 选项=值,...,.../> #定义数据节点

name #数据节点名

datahost #数据库服务器主机名

database #数据库名

<dataNode name="dn1" dataHost="mysql53" database="db1" />

<dataNode name="dn2" dataHost="mysql54" database="db2" />

<dataNode name="dn3" dataHost="mysql55" database="db3" />

主机名 dataHost 和库名 database 要一一对应

定义数据库服务器 IP 地址及端口

<datahost 选项=值,...>...</datahost> #服务器主机名

name #主机名,与 datahost 对应的主机名

host #主机名,与 IP 地址对应的主机名

url #数据库服务 IP 地址及端口号

user #数据库服务器授权用户

password #授权用户密码

<dataHost name="mysql53" maxCon="1000" minCon="10" balance="0"

writeType="0" dbType="mysql" dbDriver="native"

switchType="1" slaveThreshold="100">

名称要一  
一对应



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

```
<writeHost host="host53" url="192.168.4.53:3306" user="admin" #user 可自定义
```

不可重复

```
password="123qqq...A"> #password 可自定义
```

```
<readHost host="host52" url="192.168.1.200:3306" user="root"
```

```
password="xxx" /> #此行删除,不定义读写分离
```

```
</writeHost>
```

```
</dataHost>
```

复制此段,定义 54 55 的数据库服务器 IP 地址和端口

### 3.6 配置数据库服务器

根据分片配置做相应的设置,添加授权用户 admin,创建存储数据库 db1 db2 db3

#### 3.6.1 授权用户

```
53 mysql> grant all on *.* to admin@%" identified by "123qqq...A";
```

```
54 mysql> grant all on *.* to admin@%" identified by "123qqq...A";
```

```
55 mysql> grant all on *.* to admin@%" identified by "123qqq...A";
```

#### 3.6.2 创建数据库

```
53 mysql> create database db1;
```

```
54 mysql> create database db2;
```

```
55 mysql> create database db3;
```

### 3.7 启动 mycat 服务

```
mysql56 ~]# /usr/local/mycat/bin/mycat -help #查看命令用法
```

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

mysql56 ~]# /usr/local/mycat/bin/mycat status #查看服务状态

Mycat-server is not running.

mysql56 ~]# /usr/local/mycat/bin/mycat start #启动服务

Starting Mycat-server...

mysql56 ~]# /usr/local/mycat/bin/mycat status #查看服务状态

Mycat-server is running (23569).

mysql56 ~]# ps -C java #查看 Java 进程

PID	TTY	TIME	CMD
23571	?	00:00:03	java

mysql56 ~]# netstat -antulp | grep :8066 #查看 Java 端口

tcp6	0	0	:::8066	:::*	LISTEN	23571/java
------	---	---	---------	------	--------	------------

mysql56 ~]# ls /usr/local/mycat/logs/ #查看 mycat 日志

mycat.log mycat.pid wrapper.log

### 3.8 客户端连接及查看

客户端 50 连接分片服务器 56 访问\存取数据

mysql50 ~]# ping -c 4 192.168.4.56 #测试 50 与 56 的连接

mysql50 ~]# which mysql #检查 50 是否安装 mysql

mysql50 ~]# mysql -h192.168.4.56 -P8066 -uroot -p123456;

#用户名和密码在 56 上的/usr/local/mycat/conf/server.xml 中定义

```
50 mysql> show databases;
```

```
| DATABASE |
```

```
| TESTDB   |      #显示在 56 上的 server.xml 中定义的虚拟库
```

### 3.9 选择库

```
50 mysql> use TESTDB;
```

```
50 mysql> show tables; #显示在 schema.xml 中定义的虚拟表
```

```
50 mysql> desc company; #查看任意一张虚拟表,提示不存在
```

```
ERROR 1146 (42S02): Table 'db1.company' doesn't exist
```

### 3.10 分片规则

**sharding-by-intfile 枚举法 mod-long 求模法**

#### 3.11 sharding-by-intfile 枚举法

字段值必须在列举范围内选择

##### 3.11.1 查看 schema.xml 表中 employee 的 rule 为 sharding-by-intfile

```
@mysql56 ~]# vim /usr/local/mycat/conf/schema.xml
```

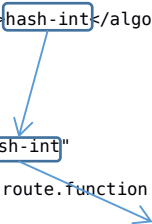
```
<table name="employee" primaryKey="ID" dataNode="dn1,dn2,dn3"
      rule="sharding-by-intfile" />
```

##### 3.11.2 查看 rule.xml 表,找到该 rule 对应的算法,再找到该算法的配置文件

```
@mysql56 ~]# vim /usr/local/mycat/conf/rule.xml
```

```
26 <tableRule name="sharding-by-intfile">
```

```
27 <rule>
28     <columns>sharding_id</columns> #employee 表中必须有此
29     <algorithm>hash-int</algorithm>
30 </rule>
31 </tableRule>
97 <function name="hash-int"
98     class="io.mycat.route.function.PartitionByFileMap">
99     <property name="mapFile">partition-hash-int.txt</property>
100 </function>
```



### 3.11.3 查看算法的配置文件 partition-hash-int.txt, 在文件内定义枚举值

```
mysql56 conf]# vim /usr/local/mycat/conf/partition-hash-int.txt
```

```
10000=0    #数据存储到 dn1 库里
10010=1    #数据存储到 dn2 库里
10020=2    #添加此行, 让数据存储到 dn3 库里
```

**#employee 表中的 sharding\_id 字段必须在这 3 个内选, 枚举法**

### 3.11.4 重启服务

```
mysql56 ~]# /usr/local/mycat/bin/mycat stop #关闭服务
```

Stopping Mycat-server...

```
mysql56 ~]# /usr/local/mycat/bin/mycat start #开启服务
```

Starting Mycat-server...

```
mysql56 ~]# netstat -antulp | grep :8066    #查看端口
```

```
tcp6  0  0  :::8066      :::*      LISTEN    23995/java
```

### 3.11.5 建表

表名在 schema.xml 已定义好, schema.xml 文件中定义了主键 ID 字段

```
<table name="employee" primaryKey="ID" dataNode="dn1,dn2,dn3"
        rule="sharding-by-intfile" />
```

rule.xml 文件中定义了枚举型 sharding\_id 字段

```
<columns>sharding_id</columns>
```

以上两个字段在 employee 表中必须存在,且 ID 为主键,sharding\_id 为枚举字段

### 3.11.6 50 连接 56 的数据库,在虚拟库 TESTDB 中创建 employee 表

```
mysql50 ~]# mysql -h192.168.4.56 -uroot -p123456 -P8066
```

```
50 mysql> use TESTDB;
```

```
50 mysql> create table TESTDB.employee(
```

```
-> ID int primary key auto_increment,
```

```
-> sharding_id int,
```

```
-> name char(15),
```

```
-> age tinyint,
```

```
-> sex enum("man","woman"));
```

### 3.11.7 在 53 54 55 上查看此表是否存在

```
mysql53 ~]# mysql -uroot -p123456
```

53 mysql> show databases; 有 db1 库

53 mysql> use db1;

53 mysql> show tables; 有 employee 表

53 mysql> desc employee;

54 在 db2 55 在 db3 中查看表 employee

### 3.11.8 50 连接 56 数据库,并插入记录

50 mysql> insert into employee(sharding\_id,name,age,sex)  
values(10000,"tom1",19,"man");

50 mysql> insert into employee(sharding\_id,name,age,sex)  
values(10010,"tom2",19,"man");

50 mysql> insert into employee(sharding\_id,name,age,sex)  
values(10020,"tom3",19,"man");

50 mysql> select \* from employee;

ID	sharding_id	name	age	sex
1	10010	tom2	19	man
1	10000	tom1	19	man
1	10020	tom3	19	man

### 3.11.9 53 54 55 上查看数据库 TESTDB.employee 内的值

53 mysql> select \* from employee; #只有 10000 这行的表记录

54 mysql> select \* from employee; #只有 10010 这行的表记录

```
55 mysql> select * from employee; #只有 10020 这行的表记录
```

以上设置达到了按枚举法分片存储数据,但客户机 50 上能访问到所有数据

### 3.12 mod-long 求模法

根据字段值与设定的数字求模结果存储数据

#### 3.12.1 查看 schema.xml 表中 hotnews 的 rule 为 mod-long

```
mysql56 ~]# vim /usr/local/mycat/conf/schema.xml
```

```
<table name="hotnews" primaryKey="ID" autoIncrement="true"  
dataNode="dn1,dn2,dn3" rule="mod-long" />
```

下划线部分删除,不使用主键 ID 及自增长

#### 3.12.2 查看 rule.xml 表,找到该 rule 对应的算法,再找到该算法

```
mysql56 ~]# vim /usr/local/mycat/conf/rule.xml
```

```
<tableRule name="mod-long">
```

```
    <rule>
```

```
        <columns>id</columns>    #建表必须的字段
```

```
        <algorithm>mod-long</algorithm>
```

```
    </rule>
```

```
</tableRule>
```

```
<function name="mod-long" class="io.mycat.route.function.  
PartitionByMod">
```

```
    <property name="count">3</property>
```

</function> #对 3 取余,3 为数据库服务器数量

### 3.12.3 重启服务

```
mysql56 ~]# /usr/local/mycat/bin/mycat stop
```

Stopping Mycat-server...

```
mysql56 ~]# /usr/local/mycat/bin/mycat start
```

Starting Mycat-server...

### 3.12.4 50 连接 56 的数据库,在虚拟库 TESTDB 中创建 hotnews 表

```
mysql50 ~]# mysql -h192.168.4.56 -uroot -p123456 -P8066
```

```
50 mysql> create table hotnews(
```

```
-> id int,
```

```
-> title char(50),
```

```
-> worker char(10),
```

```
-> comment varchar(150));
```

```
53 mysql> desc hotnews;
```

```
54 mysql> desc hotnews;
```

```
55 mysql> desc hotnews;
```

```
50 mysql> insert into hotnews(id,title,worker,comment)
```

```
-> values(8,"linux","redhat","linux-lb");
```

```
50 mysql> insert into hotnews(id,title,worker,comment)
```

```
-> values(9,"linux","redhat","linux-lb9");
```



```
50 mysql> insert into hotnews(id,title,worker,comment)
-> values(10,"linux","redhat","linux-lb10");

53 mysql> select * from hotnews; #显示 id 为 9 的行的表记录

54 mysql> select * from hotnews; #显示 id 为 10 的行的表记录

55 mysql> select * from hotnews; #显示 id 为 8 的行的表记录

50 mysql> select * from hotnews; #显示所有内容
```

### 3.13 数据不分片存储

#### 3.13.1 查看数据不分片存储的内容

```
mysql56 ~]# vim /usr/local/mycat/conf/schema.xml

<table name="company" primaryKey="ID" type="global"
dataNode="dn1,dn2,dn3" />

<table name="goods" primaryKey="ID" type="global"
dataNode="dn1,dn2,dn3" />
```

#### 3.13.2 未修改配置文件,不重启服务

#### 3.13.3 50 连接 56 的数据库,在虚拟库 TESTDB 中创建 goods 表

```
mysql50 ~]# mysql -h192.168.4.56 -uroot -p123456 -P8066

50 mysql> create table goods(ID int primary key,name
char(15),class char(15));
```

#### 3.13.4 53 54 55 上查看此表是否存在

#### 3.13.5 50 连接 56 的数据库,在 goods 表中写入值

```

50      mysql>      insert      into      goods      (ID,name,class)
values(1,"tom","nsd1906");

50      mysql>      insert      into      goods      (ID,name,class)
values(2,"jim","nsd1906");

50      mysql>      insert      into      goods      (ID,name,class)
values(3,"max","nsd1906");

```

3.13.6 53 54 55 上查看 goods 表内的值

```

53 53 55  mysql> select * from goods;  #3 台数据服务器显示内容一致

```

## 四 添加新库新表

### 4.1 添加新库

#### 修改 server.xml 文件

```

mysql56 ~]# vim /usr/local/mycat/conf/server.xml

```

```

<user name="root">

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

    <property name="schemas">TESTDB, BBSDB</property>

</user>    #在这里添加新库名

<user name="user">    #user 对 BBSDB 库只能读

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

    <property name="schemas">TESTDB, BBSDB</property>

    <property name="readOnly">true</property>

```

```
</user>
```

## 4.2 添加新表

### 修改 schema.xml 文件

```
mysql56 ~]# vim /usr/local/mycat/conf/schema.xml
```

复制<schema>.....</schema>段,粘贴,修改

```
<schema name="BBSDB" checkSQLschema="false" sqlMaxLimit="100">
    <table name="company" primaryKey="ID" type="global"
    dataNode="dn1,dn2,dn3" />
    <table name="employee" primaryKey="ID"
    dataNode="dn1,dn2,dn3" rule="sharding-by-intfile" />
    <table name="hotnews" dataNode="dn1,dn2,dn3"
    rule="mod-long" />
</schema>
```

## 4.3 重启服务

```
mysql56 ~]# /usr/local/mycat/bin/mycat stop
```

```
mysql56 ~]# /usr/local/mycat/bin/mycat start
```

```
mysql56 ~]# netstat -antulp | grep :8066
```

```
tcp6  0  0  :::8066  :::*  LISTEN  24430/java
```

## 4.4 测试配置

### 4.4.1 50 连接 56 的数据库,查看虚拟库 BBSDB 和里面的表

```
50 mysql> use BBSDB;
```

```
50 mysql> show tables; #有 company2 employee2 hotnews2 3 个表
```

4.4.2 同三中步骤在虚拟库 BBSDB 中创建表和插入值,并在 53 54 55 上查看.