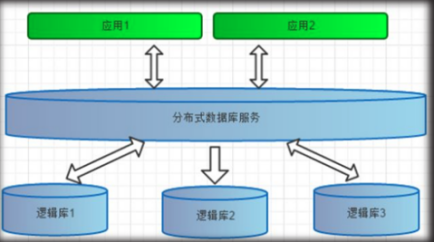
**10\_RDMBS203数据分片概述部署mycat服务测试配置**

**一 数据分片概述**

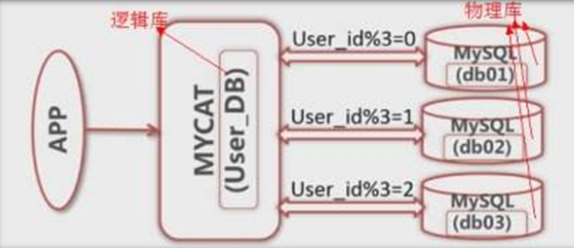
**1.1 分库/分表**



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

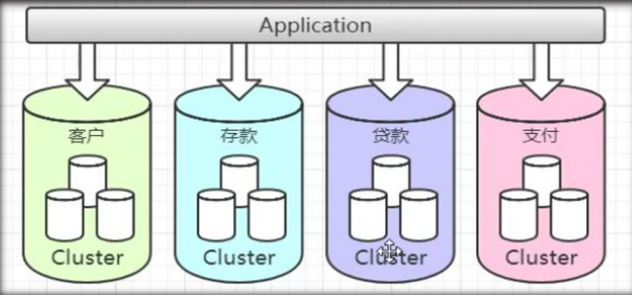
**1.2 分隔方式-水平分隔**

**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**

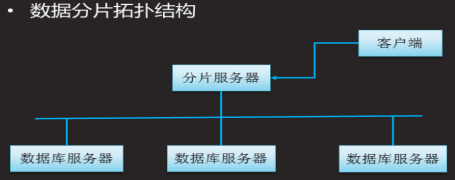
当mycat收到一个SQL命令时

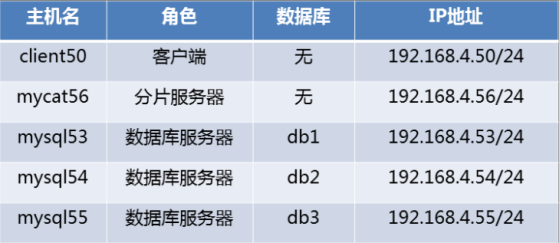
2.3.1解析SQL命令涉及到的表

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.





**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](mailto: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

**名称要一**

**一对应**

host #主机名,与IP地址对应的主机名

url #数据库服务IP地址及端口好

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">

<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="hostS2" 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 deployee;

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**

ysql56 ~]# 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.