mysql\_day08数据分片

1. 相关概念

1.1分库/分表

定义:将存放在一台数据库服务器中的数据,按照特定方式进行拆分,分散存放到多台数据库服务中,以达到分散

单台服务器负载的效果.

1.2按业务类型可以分为:水平分割和纵向分割

横向分割:按照表中指定字段的分片规则,将表记录按行切片,分散存储到所数据表中

纵向分割:将单个数据库的多个表按业务类型分类,分散存储到不通数据库

1. MyCat软件介绍

2.1软件介绍

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式存储提供解决方案

\*适合数据大量写入和存储需求

\*支持mysql,Oracle,Sqlserver,Mongodb等

\*提供数据读写分离服务

\*基于阿里巴巴Cobar进行研发的开源软件

2.2分片规则

mycat支持提供10种分片规则

[1]枚举法 sharding-by-intdile [6]通配取模 sharding-by-pattern

[2]固定分片 rule1 [7] ASCII码求模通配 sharding-by-prefixpattern

[3]范围约定 auto-sharding-long [8]变成指定 sharding-by-substring

[4]求模法 mod-long [9]字符串拆分hash解析 sharding-by-murmur

[5]日期列分区法 sharding-by-date [10]一致性hash sharding-by-murmu r

2.3挂牌工作过程

当mycat收到sql命令时

[1]解析SQL命令涉及到的表

[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,获取分片列表

[3]然后将SQL命令发往对应的分片服务器去执行

[4]最后收集和处理所有分片结果数据,并返回到客户端

1. 配置数据分片

3.1配置数据分片服务器192.168.4.56

3.1.1 安装软件

[root@host56 ~]# yum -y install java-1.8.0-openjdk //安装jdk

[root@host56 ~]# tar -zxvf Mycat-server-1.6-RELEASE-20161028204710-linux.tar.gz /解压即可用

[root@host56 ~]# mv mycat/ /usr/local/

3.1.2 配置文件说明

ls /usr/local/mycat

bin //mycat命令

catlet //扩展功能

conf //配置文件

lib //mycat使用的jar包

logs //mycat启动日志和运行日志

wrapper.log //mycat服务启动日志

mycat.log //记录sql脚本执行后的报错内容

3.1.3 修改配置文件

重要配置文件说明

--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

3.1.3.1 配置客户端连接使用的用户即面膜

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

3.1.3.2 配置数据分片

[root@host56 conf]# sed -i '56,77d' schema.xml #删除56-77行

[root@host56 conf]# sed -i '39,42d' schema.xml #删除39-42行

[root@host56 conf]# sed -i '16,18d' schema.xml #删除16-18行

[root@host56 conf]# sed -i '45,47d' schema.xml #删除45-47行

[root@host56 conf]# wc -l schema.xml

45 schema.xml

定义数据分片的表

--<schema>......</schema> //定义分片信息

--<table>......</table> //定义表

--name //逻辑库名或逻辑表明

--dataNode //指定数据库节点

--rule //指定使用的分片规则

--type=globle //数据不分片存储

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

rule="sharding-by-intfile">

</schema>

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

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

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

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

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 //数据库服务器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">

<heartbeat>select user()</heartbeat>

<writeHost host="hostM1" url="192.168.4.53:3306" user="admin"

password="123qqq...A">

</writeHost>

</dataHost>

3.2 配置数据库服务器192.168.4.53/54/55(用户授权及创建数据库)

[root@host53 ~]# mysql -u root -p123qqq...A

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

mysql> create database db1;

[root@host54 ~]# mysql -u root -p123qqq...A

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

mysql> create database db2;

[root@host55 ~]# mysql -u root -p123qqq...A

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

mysql> create database db3;

1. 启动服务

[root@host56~]#cd /usr/local/mycat/bin

[[root@host56~]#./mycat](mailto:[root@host56]#./mycat) --help

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

[root@host56~]#/usr/local/mycat/bin/mycat start

Starting Mycat-server...

Removed stale pid file:/usr/local/mycat/logs/mycat.pid

1. 查看服务信息

查看进程

[root@host56 conf]# ps -C java

查看端口

[root@host56 conf]# netstat -utnlp | grep :8066

日志文件

[root@host56 conf]# ls /usr/local/mycat/logs/

1. 客户端50连接分片服务器56访问数据

[root@host50 ~]# mysql -h192.168.4.56 -P8066 -uroot -p123456

mysql> showdatabases;

mysql> use TESTDB

mysql> show tables;

mysql> desc goods; #并不是真实存在的表

MySQL> exit

1. 分片规则使用

枚举法(sharding-by-intfile)

--字段值必须在列表范围内选择

[root@host56 ~]# vim /usr/local/mycat/conf/schema.xml

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

rule=”sharding-by-intfile”/>

[root@host56 ~]# vim /usr/local/mycat/conf/rule.xml

<tableRule name=”sharding-by-intfile”>

<rule>

<columns>sharding\_id</columns>

<algorithm>hash-int</algorithm>

<rule>

</tableRule>

<function name=”hash-int”

class=”io.mycat.route.function.PartitionByFileMap”>

<property name=”mapFile>partition-hash-int.txt</proerty>

</function>

[root@host56]#vim /usr/local/mycat/conf/partition-hash-int.txt

10000=0 //数据存储在dn1库里

10010=1 //数据存储到dn2库里

10020=2 //数据存储到dn3库里

:wq

[root@host56 ~]# /usr/local/mycat/bin/mycat stop

[root@host56 ~]# /usr/local/mycat/bin/mycat start

求余法(mod-long)

--根据字段值与设定的数字求模结果存储数据

[root@host56 ~]# vim /usr/local/mycat/conf/schema.xml

<table name="hotnews" dataNode="dn1,dn2,dn3" rule="mod-long" />

[root@host56 ~]# vim /usr/local/mycat/conf/rule.xml

<tableRule name="mod-long"> #定义表规则

<rule> #规则内容

<columns>id</columns> #字段名称

<algorithm>mod-long</algorithm> #定义求余规则

</rule>

</tableRule>

<function name="mod-long" class="io.mycat.route.function.PartitionByMod"> #取余法程序位置

<!-- how many data nodes -->

<property name="count">3</property> #与3取余

</function>

无分片规则

[root@host56 ~]# vim /usr/local/mycat/conf/schema.xml

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

创建表并测试

枚举法添加新库表

[root@host50]#mysql -h192.168.4.56 -P8066 -uroot -p123456

mysql> create table employee (ID int primary key auto\_increment,sharding\_id int,name

char(15),age tinyint,sex enum("man","woman")); #创建表

mysql> desc employee; #查看表结构

mysql> insert into employee(sharding\_id,name,age,sex) values(10010,"bob",21,"man");

mysql> insert into employee(sharding\_id,name,age,sex) values(10020,"bob",21,"man");

mysql> insert into employee(sharding\_id,name,age,sex) values(10000,"haiku",21,"man");

求余法添加新库表

[root@host50]#mysql -h192.168.4.56 -P8066 -uroot -p123456

mysql> create table hotnews(id int,title char(50),author char(10), commen varchar(150));

mysql> insert into hotnews(id,title,author,commen) values(8,"linux","redhat","linux-lb");

mysql> insert into hotnews(id,title,author,commen) values(8,"redtts","redhat","linux-lb");

mysql> insert into hotnews(id,title,author,commen) values(6,"shell","redhat","linux-lb");

无分片规则添加新库表

[root@host50]#mysql -h192.168.4.56 -P8066 -uroot -p123456

mysql> create table goods(ID int primary key,name char(15),class char(9));

mysql> insert into goods(ID,name,class) values(1,"bob","NSD1906");

mysql> insert into goods(ID,name,class) values(3,"jick","NSD1906");

mysql> insert into goods(ID,name,class) values(2,"tob","NSD1906");

1. 添加新库/新表

8.1 添加新库

[root@host56]#vim /usr/local/mycat/cong/server.xml

<user name="root">

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

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

</user>

<user name="user">

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

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

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

</user>

[root@host56]#vim /usr/local/mycat/cong/schema.xml

<schema name="BBSDB" checkSQLschema="false" sqlMaxLimit="100">

<table name="company2" primaryKey="ID" type="global" dataNode="dn1,dn2,dn3" />

<table name="employee2" primaryKey="ID" dataNode="dn1,dn2,dn3"

rule="sharding-by-intfile" />

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

dataNode="dn1,dn2,dn3" rule="mod-long" />

</schema>

[root@host56 ~]# /usr/local/mycat/bin/mycat stop #停止服务

[root@host56 ~]# /usr/local/mycat/bin/mycat start #重启服务

8.2 客户端50机器测试

[root@host50 ~]# mysql -h192.168.4.56 -P8066 -uroot -p123456

mysql>show databases;

+--------------+

| DATABASE |

+---------------+

| BBSDB |

| TESTDB |

+--------------+

mysql> create table hotnews2(num int primary key auto\_increment,name char(5));