#############################################################################################

**分片储存**

**相关概念**

垂直分割: 单个数据库的多个表,按业务类型分类,分散到不同的数据库

水平分割: a:5000 b:5000 c:5000

中间件: 数据库服务器和客户端之间的

逻辑库: 客户端视角看到的库,表的名字(假库的名字)

为高并发环境的分布式储存提供解决方案

分散单台服务器负载的效果

**分片服务器**

MyCat

*Mycat是基于java的(就是需要java环境)分布式数据库系统中间件,开源,前身:cobar(哭吧)*

适合数据大量写入的储存需求(因分布式储存有多个数据库,查找慢)

支持提供10种分片规则

可做读写分离(本实验不用)

工作原理:

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名利里分片字段值,并匹配分片函数

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

*先解析命令涉及到的表用不用分片,用则获取命令里规定的特殊值,以特殊值来决定把此命令分配到哪个服务器执行*

*实质: 命令转发*

**端口: 8066**

**文件:**

/usr/local/:

conf(配置文件):

\* .xml为配置文件 ??????????????

rule.xml 分片规则文件

\* .txt 分片规则的配置文件

\* .properties 分片规则的配置文件

lib(库文件,不能动): java程序写的压缩包文件,运行时要调用的java程序

version.txt(版本信息): 对软件的介绍

Logs(日志):

wrapper.log (服务启动日志)

mycat.log 记录mysql脚本执行后的报错内容

tail -f 日志目录

**重要文件:**

**server.xml 设置连接账号及能看到的逻辑库**

**schema.xml 配置数据分片,后端**

**rule.xml 分片规则**

**其他文件 函数调用文件**

分片规则(算法)

############################################################################

部署mycat

分片服务器: 192.168.4.56

1. 配置java环境

]#yum -y install java-1.8.0-openjdk

]#which java #查看java

]#java -version #查看java版本

1. 装包

-- 把包解压后就可以用了

]$ scp /linux-soft/03/mysql/Mycat-server-1.6-RELEASE-20161028204710-

linux.tar.gz root@192.168.4.56:/root/

]#tar -xf Mycat-server-1.6-RELEASE-20161028204710-linux.tar.gz

#解压完成后会多出mycat目录,将其移动到/usr/local/下

]#mv mycat /usr/local/

目录结构



**xml:**网页语言,标记成对存在,单标签,双标签

注释格式: <!-- -->

1. **修改配置文件**

**3.1定义连接用户和逻辑库(**server.xml **)**

]# vim /usr/local/mycat/conf/server.xml (使用默认配置)

80 <user name="root"> #客户端连接使用的用户名

81 <property name="password">密码</property>

82 <property name="schemas">TESTDB,GAMEDB</property>

登录后能看到的假库名

84 </user>

**3.2.数据分片配置(**schema.xml)

文件内容多先做备份

]# cp schema.xml /root/

删除与实验无关的内容

]# wc -l schema.xml

77 schema.xml #初始行数:77

]# sed -i '56,77d' schema.xml

]# sed -i '39,42d' schema.xml #删除后的39-42!!!

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

47 45-46 43 42 31-32 28-29 #每个页数都是删除后的页数!!!

]# wc -l schema.xml

40 schema.xml #删完后行数:40行

修改配置

库 -->表,db1 dn1 ---> local53 ---> hostM1,IP,用户

节点名 主机名

---------------------------------------------------------------------------------------------------

#定义假库(TESTDB)下的表及表分片方式(一个<table 代表一个表)

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

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

表名 储存数据的节点名 分片规则

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

.............

</schema>

------------------------------------------------------------------------------------------------------

#定义dnX(节点名)对应的主机名(dnX ---> localhostXX)

29 <dataNode name="dn1" dataHost="localhost53" database="db1" />

节点名 节点的主机名 用db1库储存

30 .......

------------------------------------------------------------------------------------------------------------

#定义localhostXX对应的IP地址, 分片服务器连接此主机myslq的用户

32 <dataHost name="localhost53" maxCon="1000" minCon="10" balance="0"

33 ............. ............... ..............

36 <writeHost host="hostM1" url="192.168.4.53:3306" user="pljyaya"

37 password="123qqq...A">

38 </writeHost>

39 </dataHost>

------------------------------------------------------------------------------------------------------------

**4.配置数据库服务器**

51-53操作:

创建配置文件中对应的库:

> create database db1;

.........

授权对应用户:

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

**5.启动服务**

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

Starting Mycat-server... #不要看这条,看端口判断是否启动成功

]#ss -antup |grep 8066

1. **查看服务状态**
2. **分片规则**

1.枚举法

Sharding-by-intfile

默认10000

都要赋值

2.求模法

Mod-long

Id % 3

删文件中的主键

Insert 必须加字段表

3.不分片

global

在表设置里添加: type="global"

server.xml ---> 配置假库,客户端连接的用户

scschema.xml ---> 删内容, 假库的表,分片,节点ip,连接用户