

# NSD DBA2 DAY05

## 1. 案例1：搭建mycat 分片服务器

### 1 案例1：搭建mycat 分片服务器

#### 1.1 问题

- 数据库主机 192.168.4.55 使用db1库存储数据
- 数据库主机 192.168.4.56 使用db2库存储数据
- 主机 192.168.4.54 运行mycat服务，逻辑库名称为test，连接用户名为admin，密码123456
- 在主机 192.168.4.254 访问测试配置

1.

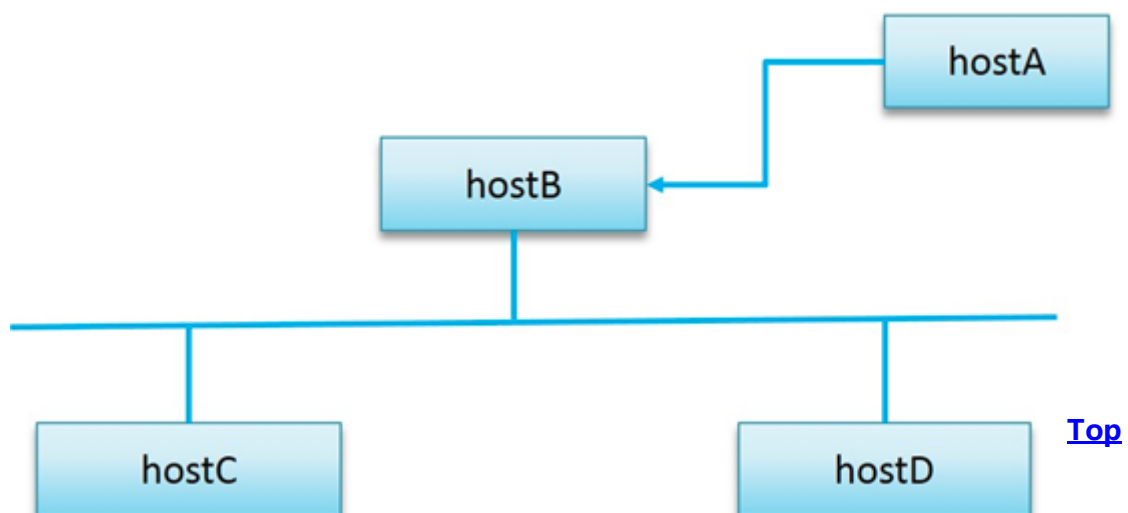
#### 1.2 方案

准备四台主机，搭建mycat分片服务器，通过某种特定条件，将存放在一个数据库(主机)中的数据，分散存放到多个数据库(主机)中，已达到分散单台设备负载的效果。其中192.168.4.56作为mycat服务器，192.168.4.54和192.168.4.55作为数据库服务器，192.168.4.254作为客户端。如图-1所示：

拓扑名称	主机名	角色	数据库名	IP地址
hostA	client	客户端	无	192.168.4.254/24
hostB	mycat	mycat服务器	无	192.168.4.56/24
hostC	c1	数据库服务器	db1	192.168.4.55/24
hostD	c2	数据库服务器	db2	192.168.4.54/24

图-1

数据分片的拓扑如图-2所示：



## 1.3 步骤

实现此案例需要按照如下步骤进行。

### 步骤一：公共配置

- 1) 关闭防火墙和selinux，配置yum源（系统源），这里不再操作
- 2) 把54，55还原成独立数据库服务器，且只保留默认4个库其他都删除，停止56主机的mha管理服务，下载mycat 软件到56主机
- 3) 在54（主机c2）和55（主机c1）上面创建db1和db2库

```

01. [root@c1 ~]# mysql -u root -p123456
02. mysql> create database db1;           //c1上面创建db1库
03. Query OK, 1 row affected (0.00 sec)
04.
05. [root@c2 ~]# mysql -u root -p123456
06. mysql> create database db2;           //c2上面创建db2库
07. Query OK, 1 row affected (0.00 sec)

```

在54上面授权一个用户

```

01. mysql> grant all on *.* to admin@ "%" identified by "123456";
02. Query OK, 0 rows affected, 1 warning (0.00 sec)

```

在55上面授权一个用户

```

01. mysql> grant all on *.* to admin@ "%" identified by "123456";
02. Query OK, 0 rows affected, 1 warning (0.00 sec)

```

### 4) 修改数据库的配置文件

注意：1代表不区分表名的大小写，0为区分大小写

主机c1上面：

```

01. [root@c1 ~]# vim /etc/my.cnf
02. [mysqld]
03. lower_case_table_names=1           //表名忽略大小写
04. [root@c1 ~]# systemctl restart mysqld

```

[Top](#)

主机c2上面：

```
01. [ root@c2 ~] # vim /etc/my.cnf
02. [ mysql]
03. lower_case_table_names=1
04. [ root@c2 ~] # systemctl restart mysqld
05.
```

5) 在56主机上面安装JDK

```
01. [ root@my cat ~] # rpm - qa | grep - ijdk //安装自带的即可
02. java-1.8.0-openjdk-1.8.0.131-11.b12.el7.x86_64
03. java-1.8.0-openjdk-headless-1.8.0.131-11.b12.el7.x86_64
04. copy-jdk-configs-2.2-3.el7.noarch
05. [ root@my cat ~] # yum - y install java-1.8.0-openjdk
```

6) 在56主机上面安装mycat

```
01. [ root@my cat ~] # cd mysql/
02. [ root@my cat mysql] # tar - xf My cat- server- 1.4 beta- 20150604171601- linux.tar.gz //免
03. [ root@my cat mysql] # mv my cat/ /usr/local/
04. [ root@my cat mysql] # ls /usr/local/my cat/
05. bin catlet conf lib logs version.txt
06. [ root@my cat mysql] # cd /usr/local/my cat/
07. [ root@my cat my cat] # ./bin/my cat -- help
08. Usage: ./bin/my cat { console | start | stop | restart | status | dump }
```

7) 修改配置文件

目录结构说明：

bin mycat命令，如启动停止等

catlet 扩展功能

conf 配置文件

lib mycat使用的jar

log mycat启动日志和运行日志

wrapper.log mycat服务启动日志

[Top](#)

mycat.log 记录SQL脚本执行后的报错内容

重要配置文件说明：

server.xml 设置连mycat的账号信息

schema.xml 配置mycat的真实库表

rule.xml 定义mycat分片规则

配置标签说明

<user>...</user> 定义连mycat用户信息

<datanode>...</datanode> 指定数据节点

<datahost>...</datahost> 指定数据库地址及用户信息

查看server.xml配置文件

```

01. [ root@my cat my cat ] # cd conf /
02. [ root@my cat conf ] # vim server.xml
03.     </system>
04.     <user name="test">           //连接my cat服务时使用的用户名 test
05.         <property name="password">test</property>
06.     //使用test用户连接my cat用户时使用的密码
07.         <property name="schemas">TEST DB</property>
08.     //连接上my cat服务后，可以看到的库名多个时，使用逗号分隔 （是逻辑上的库名,服务
09.     </user>
10.
11.
12.     <user name="user">
13.         <property name="password">user</property>
14.         <property name="schemas">TEST DB</property>
15.         <property name="readOnly">true</property>
16.     //定义只读权限，使用定义的用户连接my cat服务后只有读记录的权限,不写这一行!
17.     </user>

```

修改schema.xml配置文件

```

01. [ root@my cat conf ] # vim schema.xml
02. <table name="travelrecord" dataNode="dn1,dn2" rule="auto- sharding- long" />
03. //travelrecord (逻辑上的，名字不能随便写，一般不动) 表分片到数据节点dn1和dn2，
04. <table name="company " primary Key ="ID" type="global" dataNode="dn1,dn2" />
05. <table name="hotnews" primary Key ="ID" dataNode="dn1,dn2" rule="mod- long" />
06. <dataNode name="dn1" dataHost="c1" database="db1" />
07. //数据节点对应的服务器 name="dn1"名称要与上面的对应 dataHost="c1"写本机主机名

```

```

08. <dataNode name="dn2" dataHost="c2" database="db2" />
09. //定义分片使用的库，所在的物理主机，真正存储数据的db1库在物理主机my sql55上
10.
11.
12.
13. 指定c1名称主机对应的ip地址
14. <dataHost name="c1" maxCon="1000" minCon="10" balance="0"
15.     writeType="0" dbType="mysql" dbDriver="native" switchType="1" slaveThresh
16.     <heartbeat>select user() </heartbeat>
17.     <!-- can have multi write hosts -->
18.     <writeHost host="c1" url="192.168.4.55:3306" user="admin"
19.         password="123456">
20. //访问数据库时，my cat服务连接数据库服务器时使用的用户名和密码
21.     <!-- can have multi read hosts -->
22.
23.     </writeHost>
24. </dataHost>
25.
26.
27. 指定c2名称主机对应的ip地址
28. <dataHost name="c2" maxCon="1000" minCon="10" balance="0"
29.     writeType="0" dbType="mysql" dbDriver="native" switchType="1" slaveThresl
30.     <heartbeat>select user() </heartbeat>
31.     <!-- can have multi write hosts -->
32.     <writeHost host="c2" url="192.168.4.54:3306" user="admin"
33.         password="123456">
34. //访问数据库时，my cat服务连接数据库服务器时使用的用户名和密码
35.     <!-- can have multi read hosts -->
36.
37.     </writeHost>
38. </dataHost>

```

## 8 ) 添加PATH路径

```

01. [ root@my cat conf ] # export PATH=/usr/local/my cat/bin:$PATH
02. [ root@my cat conf ] # echo "PATH=/usr/local/my cat/bin:$PATH" >> /etc/profile
03. [ root@my cat conf ] # source /etc/profile
04. [ root@my cat conf ] # echo $PATH
05. /usr/local/my cat/bin:/usr/local/my cat/bin:/usr/local/sbin:/usr/local/bin:/usr/

```

[Top](#)

```
06. [ root@my cat conf ] # which my cat
07. /usr/local/my cat/bin/my cat
```



### 9) 启动服务并查看端口

```
01. [ root@my cat conf ]# my cat start
02. Starting My cat- server...
03. [ root@my cat conf ]# netstat - antup | grep :8066
04. tcp6      0      0 :::8066      :::*          LISTEN      6421/java
05. [ root@my cat conf ]# ps - C java
06.      PID TTY          TIME CMD
07.    6421 ?           00:00:04 java
```

## 用admin用户登录

```

01. [ root@client ~] # mysql - h192.168.4.54 - uadmin - p123456
02. mysql> show processlist;
03. +---+-----+-----+-----+-----+-----+-----+-----+
04. | Id | User | Host                | db | Command | Time | State | Info                |
05. +---+-----+-----+-----+-----+-----+-----+-----+
06. | 3 | admin | 192.168.4.56:34580 | db2 | Sleep   | 80   |      | NULL                |
07. | 4 | admin | 192.168.4.56:34570 | db2 | Sleep   | 100  |      | NULL                |
08. | 5 | admin | 192.168.4.56:34572 | db2 | Sleep   | 40   |      | NULL                |
09. | 6 | admin | 192.168.4.56:34562 | db2 | Sleep   | 30   |      | NULL                |
10. | 7 | admin | 192.168.4.56:34564 | db2 | Sleep   | 90   |      | NULL                |
11. | 8 | admin | 192.168.4.56:34566 | db2 | Sleep   | 60   |      | NULL                |
12. | 9 | admin | 192.168.4.56:34574 | db2 | Sleep   | 70   |      | NULL                |
13. | 10 | admin | 192.168.4.56:34576 | db2 | Sleep   | 10   |      | NULL                |
14. | 11 | admin | 192.168.4.56:34578 | db2 | Sleep   | 20   |      | NULL                |
15. | 12 | admin | 192.168.4.56:34568 | db2 | Sleep   | 50   |      | NULL                |
16. | 14 | admin | 192.168.4.51:58354 | NULL | Query   | 0    | starting | show processlist |
17. +---+-----+-----+-----+-----+-----+-----+-----+
18. 11 rows in set ( 0.00 sec)
19.
20.
21.
22. [ root@client ~] # mysql - h192.168.4.55 - uadmin - p123456
23. mysql> show processlist;

```

```

24.  +---+-----+-----+-----+-----+-----+-----+-----+
25.  | Id | User | Host          | db | Command | Time | State | Info          |
26.  +---+-----+-----+-----+-----+-----+-----+-----+
27.  | 3 | root | localhost     | NULL | Sleep | 2352 |      | NULL          |
28.  | 4 | admin | 192.168.4.56:45148 | db1 | Sleep | 2 |      | NULL          |
29.  | 5 | admin | 192.168.4.56:45150 | db1 | Sleep | 62 |      | NULL          |
30.  | 6 | admin | 192.168.4.56:45160 | db1 | Sleep | 12 |      | NULL          |
31.  | 7 | admin | 192.168.4.56:45162 | db1 | Sleep | 92 |      | NULL          |
32.  | 8 | admin | 192.168.4.56:45152 | db1 | Sleep | 32 |      | NULL          |
33.  | 9 | admin | 192.168.4.56:45154 | db1 | Sleep | 42 |      | NULL          |
34.  | 10 | admin | 192.168.4.56:45156 | db1 | Sleep | 22 |      | NULL          |
35.  | 11 | admin | 192.168.4.56:45158 | db1 | Sleep | 82 |      | NULL          |
36.  | 12 | admin | 192.168.4.56:45164 | db1 | Sleep | 52 |      | NULL          |
37.  | 13 | admin | 192.168.4.56:45166 | db1 | Sleep | 72 |      | NULL          |
38.  | 14 | admin | 192.168.4.51:32796 | NULL | Query | 0 | starting | show processlist |
39.  +---+-----+-----+-----+-----+-----+-----+-----+
40.  12 rows in set ( 0.00 sec)
41.
42.  [ root@my cat conf ] # ls /usr/local/my cat/logs/
43.  my cat.log my cat.pid wrapper.log      //wrapper.log为错误日志
44.  [ root@my cat conf ] # ldconfig -v      //更新加载的模块

```

## 10 ) 客户端访问

命令：mysql -hmycat主机的IP -P端口号 -u用户 -p密码

```

01.  [ root@clent ~] # my sql - h192.168.4.56 - P8066 - utest - ptest
02.  my sql> show databases;
03.  +-----+
04.  | DATABASE |
05.  +-----+
06.  | TESTDB |
07.  +-----+
08.  1 row in set ( 0.00 sec)
09.
10.  my sql> USE TESTDB;
11.  Reading table information for completion of table and column names
12.  You can turn off this feature to get a quicker startup with - A
13.
14.  Database changed

```

[Top](#)

```

15.  my sql>
16.  my sql> show tables;
17.  +-----+
18.  | Tables in TESTDB |
19.  +-----+
20.  | company          |
21.  | customer          |
22.  | customer_addr     |
23.  | employee          |
24.  | goods             |
25.  | hotnews           |
26.  | orders            |
27.  | order_items       |
28.  | travelrecord      |
29.  +-----+
30.  9 rows in set ( 0.00 sec)

```

## 11 ) 客户端测试 :

```

01.  my sql> create table employee( id int not null primary key ,name varchar( 100) ,sharding_id i
02.  Query OK, 0 rows affected ( 0.10 sec)
03.
04.  my sql> insert into employee( id,name,sharding_id) values( 1,"bob",10000) ;
05.  Query OK, 1 row affected ( 0.03 sec)
06.
07.  my sql> insert into employee( id,name,sharding_id) values( 1,"lucy",10010) ;
08.  Query OK, 1 row affected ( 0.02 sec)
09.
10.  my sql> select * from employee;
11.  +---+-----+-----+
12.  | id | name | sharding_id |
13.  +---+-----+-----+
14.  | 1 | bob | 10000 |
15.  | 1 | lucy | 10010 |
16.  +---+-----+-----+
17.  2 rows in set ( 0.06 sec)

```

[Top](#)

## 12 ) 在c1上面查看结果



```

01.  my sql> show databases;
02.  +-----+
03.  | Database      |
04.  +-----+
05.  | information_schema |
06.  | db1           |
07.  | my sql       |
08.  | performance_schema |
09.  | sys          |
10.  +-----+
11.  5 rows in set ( 0.00 sec)
12.
13.  my sql> use db1;
14.  Reading table information for completion of table and column names
15.  You can turn off this feature to get a quicker startup with -A
16.
17.  Database changed
18.  my sql> show tables;
19.  +-----+
20.  | Tables_in_db1 |
21.  +-----+
22.  | employee      |
23.  +-----+
24.  1 row in set ( 0.00 sec)
25.
26.  my sql> select * from employee;    //查看结果
27.  +---+-----+-----+
28.  | id | name | sharding_id |
29.  +---+-----+-----+
30.  |  1 | bob  |      10000 |
31.  +---+-----+-----+
32.  1 row in set ( 0.00 sec)

```

### 13 ) 在c2上面查看结果

```

01.  my sql> show databases;
02.  +-----+
03.  | Database      |
04.  +-----+
05.  | information_schema |

```

[Top](#)

```

06. | db2          |
07. | my sql        |
08. | performance_schema |
09. | sys           |
10. +-----+
11. 5 rows in set (0.00 sec)
12.
13. my sql> use db2;
14. Reading table information for completion of table and column names
15. You can turn off this feature to get a quicker startup with -A
16.
17. Database changed
18. my sql> show tables;
19. +-----+
20. | Tables_in_db2 |
21. +-----+
22. | employee      |
23. +-----+
24. 1 row in set (0.00 sec)
25.
26. my sql> select * from employee;
27. +---+-----+-----+
28. | id | name | sharding_id |
29. +---+-----+-----+
30. | 1 | lucy | 10010 |
31. +---+-----+-----+
32. 1 row in set (0.00 sec)

```

[Top](#)