4.4 实战案例——构建读写分离的数据库集群

4.4.1 案例目标

（1）了解 Mycat 提供的读写分离功能。

（2）了解 MySQL 数据库的主从架构。

（3）构建以 Mycat 为中间件的读写分离数据库集群。

4.4.2 案例分析

1. 规划节点

使用 Mycat 作为数据库中间件服务构建读写分离的数据库集群，节点规划见表 4-4-1。

表 4-4-1 节点规划

IP	主机名	节点
172.16.51.6	mycat	Mycat 中间件服务节点
172.16.51.18	db1	MariaDB 数据库集群主节点
172.16.51.30	db2	MariaDB 数据库集群从节点

2. 基础准备

使用 CentOS 7.2 系统，flavor 使用 2vCPU/4G 内存/50G 硬盘，创建 3 台虚拟机进行实验。

其中 2 台虚拟机 db1 和 db2 部署 MariaDB 数据库服务，搭建主从数据库集群；一台作为主节点，负责写入数据库信息；另一台作为从节点，负责读取数据库信息。

使用一台虚拟机部署 Mycat 数据库中间件服务，将用户提交的读写操作识别分发给相应的数据库节点。这样将用户的访问操作、数据库的读与写操作分给 3 台主机，只有数据库集群的主节点接收增、删、改 SQL 语句，从节点接收查询语句，分担了主节点的查询压力。

Yum 源使用提供的 gpmall-repo 文件夹作为本地源，Mycat 组件使用提供的 Mycat-server-1.6-RELEASE-20161028204710-linux.tar.gz 压缩包安装。

4.4.3 案例实施

1. 基础环境配置

（1）修改主机名

使用 hostnamectl 命令修改 3 台主机的主机名。

Mycat 节点修改主机名命令：

```
[root@localhost ~]# hostnamectl set-hostname mycat
[root@localhost ~]# bash
[root@mycat ~]#
```

db1 节点修改主机名命令：

```
[root@localhost ~]# hostnamectl set-hostname db1
[root@localhost ~]# bash
[root@db1 ~]#
```

db2 节点修改主机名命令：

```
[root@localhost ~]# hostnamectl set-hostname db2
[root@localhost ~]# bash
[root@db2 ~]#
```

（2）编辑 hosts 文件

3 台集群虚拟机的/etc/hosts 文件配置部分：

```
[root@mycat ~]# cat /etc/hosts
127.0.0.1      localhost localhost.localdomain localhost4 localhost4.localdomain4
::1              localhost localhost.localdomain localhost6 localhost6.localdomain6
172.16.51.6        mycat
172.16.51.18       db1
172.16.51.30       db2
```

（3）配置 Yum 安装源

数据库集群需要安装 MariaDB 数据库服务，需要给集群虚拟机配置 Yum 安装源文件，使用提供的 gpmall-repo 文件上传至 3 个虚拟机的/opt 目录下，设置本地 Yum 源。

首先将 3 个节点/etc/yum.repo.d 目录下的文件移动到/media 下，命令如下：

```
# mv /etc/yum.repos.d/* /media/
```

3 台集群虚拟机的 Yum 安装源文件配置部分：

```
# cat /etc/yum.repos.d/local.repo
[mariadb]
name=mariadb
baseurl=file:///opt/gpmall-repo
```

```
gpgcheck=0
enabled=1
```

（4）安装 JDK 环境

部署Mycat中间件服务需要先部署JDK 1.7或以上版本的JDK软件环境，这里部署JDK 1.8版本。

Mycat 节点安装 Java 环境：

```
# yum install -y java-1.8.0-openjdk java-1.8.0-openjdk-devel
...
[root@mall ~]# java -version
openjdk version "1.8.0_222"
OpenJDK Runtime Environment (build 1.8.0_222-b10)
OpenJDK 64-Bit Server VM (build 25.222-b10, mixed mode)
```

（5）关闭防火墙（全部节点）

```
[root@mall ~]#iptables -F
[root@mall ~]#iptables -X
[root@mall ~]#iptables -Z
[root@mall ~]#iptables-save
```

2. 部署 MariaDB 主从数据库集群服务

（1）安装 MariaDB 服务

通过 YUM 命令在 db1 和 db2 虚拟机节点上安装 MariaDB 服务，命令如下：

```
# yum install -y mariadb mariadb-server
```

2 个节点启动 MariaDB 服务，并设置 MariaDB 服务为开机自启。

```
# systemctl start mariadb
# systemctl enable mariadb
```

（2）初始化 MariaDB 数据库

在 db1 和 db2 虚拟机节点上初始化 MariaDB 数据库，并设置 MariaDB 数据库 root 访问用户的密码为 123456。

```
[root@db1 ~]# mysql_secure_installation
/usr/bin/mysql_secure_installation: line 379: find_mysql_client: command not found
NOTE: RUNNING ALL PARTS OF THIS SCRIPT IS RECOMMENDED FOR ALL MariaDB
```

```
        SERVERS IN PRODUCTION USE!  PLEASE READ EACH STEP CAREFULLY!
In order to log into MariaDB to secure it, we'll need the current
password for the root user.  If you've just installed MariaDB, and
you haven't set the root password yet, the password will be blank,
so you should just press enter here.
Enter current password for root (enter for none):       #默认按回车
OK, successfully used password, moving on...
Setting the root password ensures that nobody can log into the MariaDB
root user without the proper authorisation.
Set root password? [Y/n] y
New password:                                           #输入数据库 root 密码 123456
Re-enter new password:                                  #重复输入密码 123456
Password updated successfully!
Reloading privilege tables..
 ... Success!
By default, a MariaDB installation has an anonymous user, allowing anyone
to log into MariaDB without having to have a user account created for
them.  This is intended only for testing, and to make the installation
go a bit smoother.  You should remove them before moving into a
production environment.
Remove anonymous users? [Y/n] y
 ... Success!
Normally, root should only be allowed to connect from 'localhost'.  This
ensures that someone cannot guess at the root password from the network.
Disallow root login remotely? [Y/n] n
 ... skipping.
By default, MariaDB comes with a database named 'test' that anyone can
access.  This is also intended only for testing, and should be removed
before moving into a production environment.
Remove test database and access to it? [Y/n] y
```

```
 - Dropping test database...
 ... Success!
 - Removing privileges on test database...
 ... Success!
Reloading the privilege tables will ensure that all changes made so far
will take effect immediately.
Reload privilege tables now? [Y/n] y
 ... Success!
Cleaning up...
All done!  If you've completed all of the above steps, your MariaDB
installation should now be secure.
Thanks for using MariaDB!
```

（3）配置数据库集群主节点

编辑主节点 db1 虚拟机的数据库配置文件 my.cnf，在配置文件 my.cnf 中增添下面的内容：

```
[root@db1 ~]# cat /etc/my.cnf
[mysqld]
log_bin = mysql-bin                          #记录操作日志
binlog_ignore_db = mysql                     #不同步 MySQL 系统数据库
server_id = 18                               #数据库集群中的每个节点 id 都要不同，一般使用 IP 地址的最后段的数字，例如 172.16.51.18，server_id 就写 18

datadir=/var/lib/mysql
socket=/var/lib/mysql/mysql.sock
# Disabling symbolic-links is recommended to prevent assorted security risks
symbolic-links=0

[mysqld_safe]
log-error=/var/log/mariadb/mariadb.log
pid-file=/var/run/mariadb/mariadb.pid
```

编辑完成配置文件 my.cnf 后，重启 MariaDB 服务。

```
[root@db1 ~]# systemctl restart mariadb
```

（4）开放主节点的数据库权限

在主节点 db1 虚拟机上使用 mysql 命令登录 MariaDB 数据库，授权在任何客户端机器上可以以 root 用户登录到数据库。

```
[root@db1 ~]# mysql -uroot -p123456
Welcome to the MariaDB monitor.  Commands end with ; or \g.
Your MariaDB connection id is 137
Server version: 10.3.18-MariaDB-log MariaDB Server
Copyright (c) 2000, 2018, Oracle, MariaDB Corporation Ab and others.
Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

MariaDB [(none)]> grant all privileges  on *.* to root@'%' identified by "123456";
```

在主节点 db1 数据库上创建一个 user 用户让从节点 db2 连接，并赋予从节点同步主节点数据库的权限，命令如下：

```
MariaDB [(none)]> grant replication slave on *.* to 'user'@'db2' identified by '123456';
```

（5）配置从节点 db2 同步主节点 db1

在从节点 db2 虚拟机上使用 mysql 命令登录 MariaDB 数据库，配置从节点连接主节点的连接信息。master_host 为主节点主机名 db1，master_user 为在步骤（4）中创建的用户 user，命令如下：

```
[root@db2 ~]# mysql -uroot -p123456
Welcome to the MariaDB monitor.  Commands end with ; or \g.
Your MariaDB connection id is 88
Server version: 10.3.18-MariaDB-log MariaDB Server
Copyright (c) 2000, 2018, Oracle, MariaDB Corporation Ab and others.
Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

MariaDB [(none)]> change master to
master_host='db1',master_user='user',master_password='123456';
```

配置完毕主从数据库之间的连接信息之后，开启从节点服务。使用 show slave status\G; 命令并查看从节点服务状态，如果 Slave_IO_Running 和 Slave_SQL_Running 的状态都为 YES，

则从节点服务开启成功。查询结果如图 4-4-1 所示。

```
MariaDB [(none)]> start slave;

MariaDB [(none)]> show slave status\G;
```

```
MariaDB [(none)]> show slave status\G;
*************************** 1. row ***************************
               Slave_IO_State: Waiting for master to send event
                  Master_Host: db1
                  Master_User: user
                  Master_Port: 3306
                Connect_Retry: 60
              Master_Log_File: mysql-bin.000001
          Read_Master_Log_Pos: 3815
               Relay_Log_File: db2-relay-bin.000002
                Relay_Log_Pos: 4114
        Relay_Master_Log_File: mysql-bin.000001
             Slave_IO_Running: Yes
            Slave_SQL_Running: Yes
              Replicate_Do_DB:
          Replicate_Ignore_DB: mysql
           Replicate_Do_Table:
       Replicate_Ignore_Table:
      Replicate_Wild_Do_Table:
  Replicate_Wild_Ignore_Table:
                   Last_Errno: 0
                   Last_Error:
                 Skip_Counter: 0
          Exec_Master_Log_Pos: 3815
              Relay_Log_Space: 4421
              Until_Condition: None
               Until_Log_File:
                Until_Log_Pos: 0
           Master_SSL_Allowed: No
           Master_SSL_CA_File:
           Master_SSL_CA_Path:
              Master_SSL_Cert:
            Master_SSL_Cipher:
               Master_SSL_Key:
        Seconds_Behind_Master: 0
Master_SSL_Verify_Server_Cert: No
                Last_IO_Errno: 0
                Last_IO_Error:
               Last_SQL_Errno: 0
               Last_SQL_Error:
  Replicate_Ignore_Server_Ids:
             Master_Server_Id: 18
               Master_SSL_Crl:
           Master_SSL_Crlpath:
                   Using_Gtid: No
                  Gtid_IO_Pos:
      Replicate_Do_Domain_Ids:
  Replicate_Ignore_Domain_Ids:
                Parallel_Mode: conservative
                    SQL_Delay: 0
          SQL_Remaining_Delay: NULL
      Slave_SQL_Running_State: Slave has read all relay log; waiting for the slave I/O thread to update it
           Slave_DDL_Groups: 26
Slave_Non_Transactional_Groups: 10
    Slave_Transactional_Groups: 4
1 row in set (0.000 sec)
```

图 4-4-1 查询 slave 状态

（6）验证主从数据库的同步功能

先在主节点 db1 的数据库中创建库 test，并在库 test 中创建表 company，插入表数据。创建完成后，查看表 company 数据，如下所示：

```
MariaDB [(none)]> create database test;

Query OK, 1 row affected (0.001 sec)

MariaDB [(none)]> use test

Database changed

MariaDB [test]> create table company(id int not null primary key,name varchar(50),addr varchar(255));

Query OK, 0 rows affected (0.165 sec)
```

```
MariaDB [test]> insert into company values(1,"facebook","usa");
Query OK, 1 row affected (0.062 sec)
MariaDB [test]> select * from company;
+----+----------+------+
| id | name     | addr |
+----+----------+------+
|  1 | facebook | usa  |
+----+----------+------+
1 row in set (0.000 sec)
```

这时从节点 db2 的数据库就会同步主节点数据库创建的 test 库，可以在从节点查询 test 数据库与表 company，如果可以查询到信息，就能验证主从数据库集群功能在正常运行。查询结果如下所示：

```
MariaDB [(none)]> show databases;
+--------------------+
| Database           |
+--------------------+
| information_schema |
| mysql              |
| performance_schema |
| test               |
+--------------------+
4 rows in set (0.001 sec)

MariaDB [(none)]> select * from test.company;
+----+----------+------+
| id | name     | addr |
+----+----------+------+
|  1 | facebook | usa  |
+----+----------+------+
1 row in set (0.001 sec)
```

3. 部署 Mycat 读写分离中间件服务

（1）安装 Mycat 服务

将 Mycat 服务的二进制软件包 Mycat-server-1.6-RELEASE-20161028204710-linux.tar.gz 上传到 Mycat 虚拟机的/root 目录下，并将软件包解压到/use/local 目录中。赋予解压后的 Mycat 目录权限。

```
[root@mycat ~]# tar -zxvf Mycat-server-1.6-RELEASE-20161028204710-linux.tar.gz -C /usr/local/
[root@mycat ~]# chown -R 777 /usr/local/mycat/
```

在/etc/profile 系统变量文件中添加 Mycat 服务的系统变量，并生效变量。

```
[root@mycat ~]# echo export MYCAT_HOME=/usr/local/mycat/ >> /etc/profile
[root@mycat ~]# source /etc/profile
```

（2）编辑 Mycat 的逻辑库配置文件

配置 Mycat 服务读写分离的 schema.xml 配置文件在/usr/local/mycat/conf/目录下，可以在文件中定义一个逻辑库，使用户可以通过 Mycat 服务管理该逻辑库对应的 MariaDB 数据库。在这里定义一个逻辑库 schema，name 为 USERDB；该逻辑库 USERDB 对应数据库 database 为 test（在部署主从数据库时已安装）；设置数据库写入节点为主节点 db1；设置数据库读取节点为从节点 db2。（可以直接删除原来 schema.xml 的内容，替换为如下。）

注意：IP 需要修改成实际的 IP 地址。

```
[root@mycat ~]# cat /usr/local/mycat/conf/schema.xml
<?xml version="1.0"?>
<!DOCTYPE mycat:schema SYSTEM "schema.dtd">
<mycat:schema xmlns:mycat="http://io.mycat/">
<schema name="USERDB" checkSQLschema="true" sqlMaxLimit="100" dataNode="dn1"></schema>
<dataNode name="dn1" dataHost="localhost1" database="test" />
<dataHost name="localhost1" maxCon="1000" minCon="10" balance="3" dbType="mysql" dbDriver="native" writeType="0" switchType="1"  slaveThreshold="100">
        <heartbeat>select user()</heartbeat>
        <writeHost host="hostM1" url="172.16.51.18:3306" user="root" password="123456">
            <readHost host="hostS1" url="172.16.51.30:3306" user="root" password="123456"
```

```
/>
            </writeHost>
        </dataHost>
        </mycat:schema>
```

代码说明：

- sqlMaxLimit：配置默认查询数量。
- database：为真实数据库名。
- balance="0"：不开启读写分离机制，所有读操作都发送到当前可用的 writeHost 上。
- balance="1"：全部的 readHost 与 stand by writeHost 参与 select 语句的负载均衡，简单来说，当双主双从模式（M1->S1，M2->S2，并且 M1 与 M2 互为主备），正常情况下，M2、S1、S2 都参与 select 语句的负载均衡。
- balance="2"：所有读操作都随机的在 writeHost、readhost 上分发。
- balance="3"：所有读请求随机地分发到 wiriterHost 对应的 readhost 执行，writerHost 不负担读压力，注意 balance=3 只在 1.4 及其以后版本有，1.3 版本没有。
- writeType="0"：所有写操作发送到配置的第一个 writeHost，第一个挂了需要切换到还生存的第二个 writeHost，重新启动后已切换后的为准，切换记录在配置文件 dnindex.properties 中。
- writeType="1"：所有写操作都随机的发送到配置的 writeHost。

（3）修改配置文件权限

修改 schema.xml 的用户权限，命令如下：

```
[root@mycat ~]# chown root:root /usr/local/mycat/conf/schema.xml
```

（4）编辑 mycat 的访问用户

修改/usr/local/mycat/conf/目录下的 server.xml 文件，修改 root 用户的访问密码与数据库，密码设置为 123456，访问 Mycat 的逻辑库为 USERDB，命令如下：

```
[root@mycat ~]# cat /usr/local/mycat/conf/server.xml
在配置文件的最后部分，
<user name="root">
            <property name="password">123456</property>
            <property name="schemas">USERDB</property>
```

然后删除如下几行：

```
<user name="user">
        <property name="password">user</property>
        <property name="schemas">TESTDB</property>
        <property name="readOnly">true</property>
</user>
```

保存并退出 server.xml 配置文件。

（5）启动 Mycat 服务

通过命令启动 Mycat 数据库中间件服务，启动后使用 netstat -ntpl 命令查看虚拟机端口开放情况，如果有开放 8066 和 9066 端口，则表示 Mycat 服务开启成功。端口查询情况如图 4-4-2 所示。

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

```
[root@mycat ~]# netstat  -ntpl
Active Internet connections (only servers)
Proto Recv-Q Send-Q Local Address           Foreign Address         State       PID/Program name
tcp        0      0 0.0.0.0:22              0.0.0.0:*               LISTEN      974/sshd
tcp        0      0 127.0.0.1:25            0.0.0.0:*               LISTEN      886/master
tcp        0      0 127.0.0.1:32000         0.0.0.0:*               LISTEN      14243/java
tcp6       0      0 :::8066                 :::*                    LISTEN      14243/java
tcp6       0      0 :::36290                :::*                    LISTEN      14243/java
tcp6       0      0 :::44232                :::*                    LISTEN      14243/java
tcp6       0      0 :::9066                 :::*                    LISTEN      14243/java
tcp6       0      0 :::22                   :::*                    LISTEN      974/sshd
tcp6       0      0 ::1:25                  :::*                    LISTEN      886/master
tcp6       0      0 :::1984                 :::*                    LISTEN      14243/java
```

图 4-4-2 查询 Mycat 服务端口

4. 验证数据库集群服务读写分离功能

（1）用 Mycat 服务查询数据库信息

先在 Mycat 虚拟机上使用 Yum 安装 mariadb-client 服务。

```
[root@mycat ~]# yum install -y MariaDB-client
```

在 Mycat 虚拟机上使用 mysql 命令查看 Mycat 服务的逻辑库 USERDB，因为 Mycat 的逻辑库 USERDB 对应数据库 test（在部署主从数据库时已安装），所以可以查看库中已经创建的表 company。命令如下。

```
[root@mycat ~]# mysql -h127.0.0.1 -P8066 -uroot -p123456
Welcome to the MariaDB monitor.  Commands end with ; or \g.
Your MySQL connection id is 2
Server version: 5.6.29-mycat-1.6-RELEASE-20161028204710 MyCat Server (OpenCloundDB)
```

```
Copyright (c) 2000, 2018, Oracle, MariaDB Corporation Ab and others.

Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

MySQL [(none)]> show databases;
+----------+
| DATABASE |
+----------+
| USERDB   |
+----------+
1 row in set (0.001 sec)

MySQL [(none)]> use USERDB
Reading table information for completion of table and column names
You can turn off this feature to get a quicker startup with -A

Database changed
MySQL [USERDB]> show tables;
+----------------+
| Tables_in_test |
+----------------+
| company        |
+----------------+
1 row in set (0.003 sec)
MySQL [USERDB]> select * from company;
+----+----------+------+
| id | name     | addr |
+----+----------+------+
|  1 | facebook | usa  |
+----+----------+------+
```

```
1 row in set (0.005 sec)
```

（2）用 Mycat 服务添加表数据

在 Mycat 虚拟机上使用 mysql 命令对表 company 添加一条数据(2,"basketball","usa")，添加完毕后查看表信息。命令如下。

```
MySQL [USERDB]> insert into company values(2,"bastetball","usa");
Query OK, 1 row affected (0.050 sec)


MySQL [USERDB]> select * from company;
+----+------------+------+
| id | name       | addr |
+----+------------+------+
|  1 | facebook   | usa  |
|  2 | bastetball | usa  |
+----+------------+------+
2 rows in set (0.002 sec)
```

（3）验证 Mycat 服务对数据库读写操作分离

在 Mycat 虚拟机节点使用 mysql 命令，通过 9066 端口查询对数据库读写操作的分离信息。可以看到所有的写入操作 WRITE_LOAD 数都在 db1 主数据库节点上，所有的读取操作 READ_LOAD 数都在 db2 主数据库节点上。由此可见，数据库读写操作已经分离到 db1 和 db2 节点上了。命令如下。

```
[root@mycat ~]# mysql -h127.0.0.1 -P9066 -uroot -p123456 -e 'show  @@datasource;'
```

查询结果如图 4-4-3 所示。

```
[root@mycat conf]# mysql -h127.0.0.1 -P9066 -uroot -p123456 -e 'show  @@datasource;'
+----------+--------+-------+--------------+------+------+--------+------+------+---------+-----------+------------+
| DATANODE | NAME   | TYPE  | HOST         | PORT | W/R  | ACTIVE | IDLE | SIZE | EXECUTE | READ_LOAD | WRITE_LOAD |
+----------+--------+-------+--------------+------+------+--------+------+------+---------+-----------+------------+
| dn1      | hostM1 | mysql | 172.16.51.22 | 3306 | W    |      0 |    4 | 1000 |      68 |         0 |          1 |
| dn1      | hostS1 | mysql | 172.16.51.26 | 3306 | R    |      0 |    4 | 1000 |      68 |         4 |          0 |
+----------+--------+-------+--------------+------+------+--------+------+------+---------+-----------+------------+
```

图 4-4-3 查询读写分离

至此，Mycat 读写分离数据库案例完成。