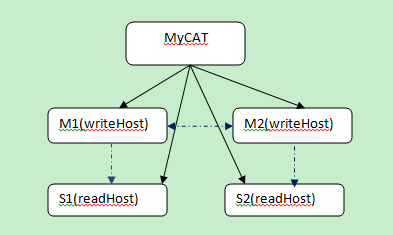
# MyCAT 高级功能测试案例

部署方式：

后端双主双从模式的数据库模式，四个实例分别命名为M1,M2,S1,S2, (M1->S1，M2->S2，并且M1与 M2互为主备)



Schema.xml配置如下：

<dataNode name=*"dn1"* dataHost=*"localhost1"* database=*"db1"* />

<dataNode name=*"dn2"* dataHost=*"localhost1"* database=*"db2"* />

<dataNode name=*"dn3"* dataHost=*"localhost1"* database=*"db3"* />

<dataHost name=*"localhost1"* maxCon=*"500"* minCon=*"10"* balance=*"1"*

dbType=*"mysql"* dbDriver=*"native"*>

<heartbeat>select user()</heartbeat>

<!-- can have multi write hosts -->

<writeHost host=*"hostM1"* url=*"localhost:3306"* user=*"root"*

password=*"123456"*>

<!-- can have multi read hosts -->

<readHost host="hostS1" url="localhost:3307" user="root" password="123456" />

</writeHost>

<writeHost host=*"hostM2"* url=*"localhost:3308"* user=*"root"*

password=*"123456"*>

<!-- can have multi read hosts -->

<readHost host="hostS2" url="localhost:3309" user="root" password="123456" />

</writeHost>

</dataHost>

**心跳功能测试：**

案例一：所有节点正常，心跳正常。

案例二：S1或S2停掉，发现检测出来心跳异常。

案例二：S1或S2恢复，发现检测出来心跳正常。

**高可用测试：**

案例一：所有节点正常，此时M1负责写数据，通过执行insert或update可以看到SQL是在M1上执行。

16:37:21.660 DEBUG [Processor0-E3] (PhysicalDBPool.java:333) -select read source hostM1 for dataHost:localhost1

观察数据库的日志或数据库本身的数据记录，可以验证这一点。

案例二：M1停止，观测到日志中发现节点切换事件，insert或update语句此时在M2上执行。

案例三：M2也停止，此时执行update 或insert语句报错，而命令行执行select语句成功。

**读写分离测试：**

环境正常，M1为当前写节点。

案例一：balance设置为0，此时，所有操作会在M1上。

案例二：balance设置为1，此时，写操作会在M1上，读操作会在M2,S2,S1上随机分配。观察日志以及查看数据库日志（数据库日志可以临时开启），可以发现这一点。

案例三：balance设置为2，此时，写操作在M1上，所有读操作会在M1 ，M2，S1，S2上随机分配。

**故障情况下的读写分离测试：**

案例一：balance为1， M1停止，此时，写操作会在M2上，读操作会在S2上

案例二：balance为1， M1停止M2停止，此时，写操作失败，读操作会在S1,S2上

案例三：balance为2， M1停止，此时，写操作会在M2上，读操作会在S2，M2上

案例四：balance为2， M1，M2停止，此时，写操作失败，读操作会在S1，S2上

案例五：balance为1， S1停止，此时，读操作会在S2、M2上 ，反之若停止S2 ，则读操作在S1、M1上。

案例六：balance为1， S1、S2都停止，则读操作在M2上。

案例七：balance为2， S1停止，则读操作在M2上，M1、S2上。

案例八：balance为2， S1，S2都停止，则读操作在M2上，M1上。