

# 32.MySQL读写分离–MHA+ProxySQL✓

## 1.ProxySQL简介

读写分离是为了在不发生故障的情况下，实现硬件利用最大化

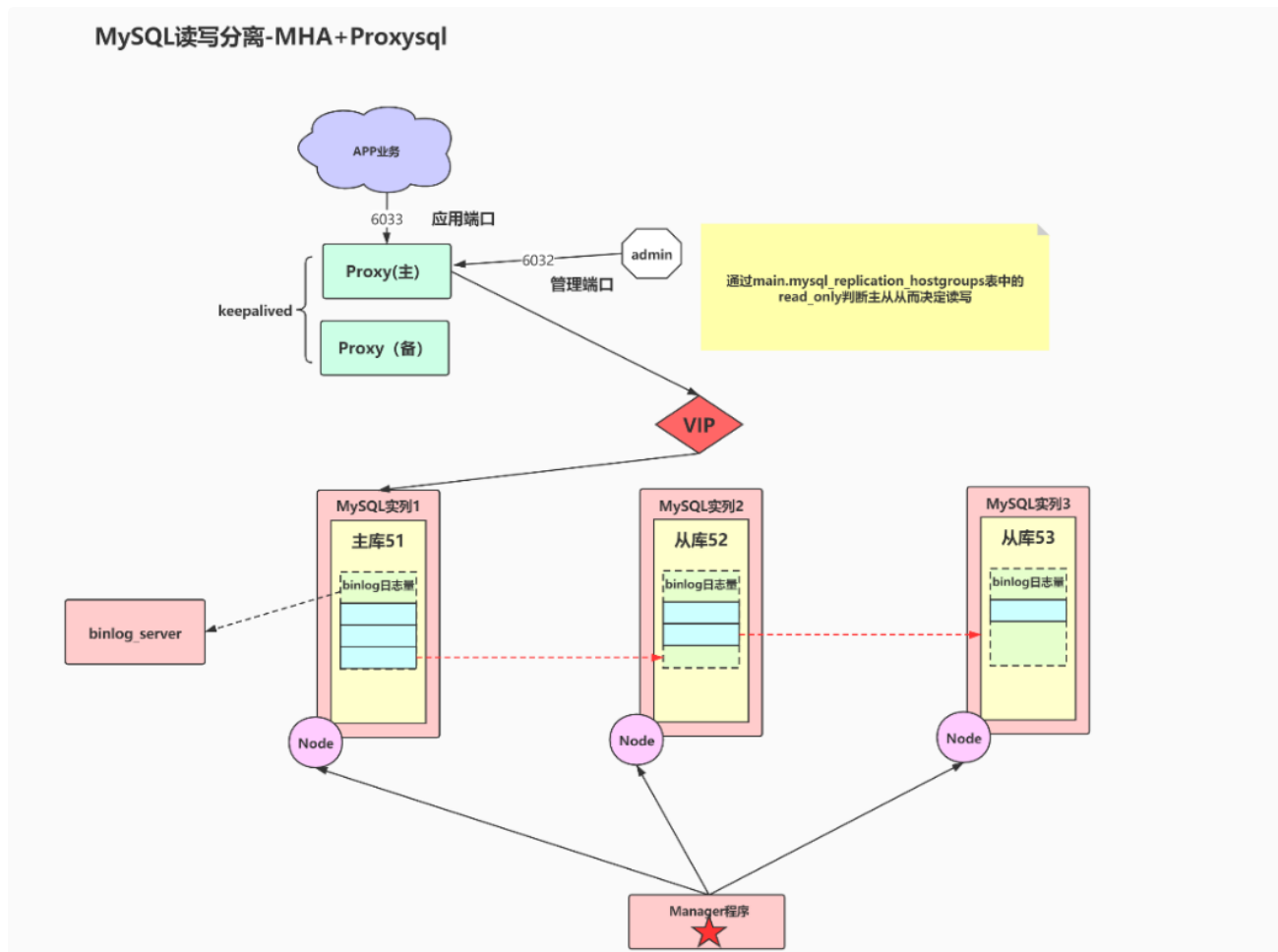
▼

Bash | Copy

```
1  MySQL官方早期开发过中间件MySQL-Proxy, 大公司一般选择自研中间件。
2  主流开源的中间件是Proxysql, 是基于MySQL-Proxy二次开发的开源产品。
3  Proxy是一个灵活的MYSQL代理层, 可以实现读写分离, 支持 Query路由功能,
4  支持动态指定某个SQL进行缓存, 支持动态加载配置信息（无需重启 ProxySQL 服务）, 支持故障切换和SQL的过滤功能。
```

Proxy相关的网站：  
<https://www.proxysql.com/> <<https://www.proxysql.com/>>  
<https://github.com/sysown/proxysql/wiki> <<https://github.com/sysown/proxysql/wiki>>

## H1 ▾ 2.MHA+ProxySQL原理图



### 3. 部署Proxy中间件

#### 3.1 安装proxySQL软件

```

1 [root@db03 ~]# rpm -ivh proxysql-2.0.10-1-centos7.x86_64.rpm
2 [root@db03 ~]# systemctl start proxysql
3 [root@db03 ~]# netstat -tulnp
4 tcp        0      0 0.0.0.0:6032        0.0.0.0:*        LISTEN      2115/proxysql
5 tcp        0      0 0.0.0.0:6033        0.0.0.0:*        LISTEN      2115/proxysql
  
```

## 3.2 管理proxySQL

### 3.2.1 登陆proxySQL

Bash | Copy

```
1  mysql -uadmin -padmin -h127.0.0.1 -P6032
2      默认用户名和密码是admin
```

### 3.2.2 查看proxySQL库结构

Bash | Copy

```
1  db03 [(none)]>show databases;
2  +-----+-----+-----+
3  | seq | name      | file                                |
4  +-----+-----+-----+
5  | 0    | main      |                                     |
6  | 2    | disk      | /var/lib/proxysql/proxysql.db      |
7  | 3    | stats     |                                     |
8  | 4    | monitor   |                                     |
9  | 5    | stats_history | /var/lib/proxysql/proxysql_stats.db |
10 +-----+-----+-----+
11
12 在ProxySQL, 6032端口共五个库: main、disk、stats 、monitor、 stats_history
13 main:
14     main库下主要的表如下
15     global_variables    关键参数
16     mysql_servers:      后端可以连接MySQL服务器的列表
17     mysql_users:        配置后端数据库的账号和监控的账号。
18     mysql_query_rules:  指定 Query 路由到后端不同服务器的规则列表(路由规则)
19     mysql_replication_hostgroups : 节点分组配置信息
20 注: 表名以 runtime_开头的表示ProxySQL 当前运行的配置内容, 不能直接修改。不带runtime_是下文图中Mem相关的配置。
21
22 disk : (SQLlie数据库进行存储)
23     持久化的磁盘的配置
24
25 stats:
26     统计信息的汇总
27
28 monitor:
29     监控的收集信息, 比如数据库的健康状态等
30
31 stats_history:
32     ProxySQL 收集的有关其内部功能的历史指标
33
```

## 3.3 配置ProxySQL

### 3.3.1 ProxySQL管理接口的多层配置关系

Bash | Copy

```
1  整套配置系统分为三层：
2  顶层    RUNTIME
3  中间层  MEMORY   （主要修改的配置表）
4  持久层  DISK  和  CFG  FILE
5
6  RUNTIME  ：
7      代表 ProxySQL 当前正在使用的配置，无法直接修改此配置，必须从下一层 （MEM层）“load” 进来。
8  MEMORY：
9      MEMORY 层上面连接 RUNTIME 层，下面disk持久层。这层可以在线操作 ProxySQL 配置，随便修改，不会影响生产环境。
10     确认正常之后在加载达到RUNTIME和持久化的磁盘上。修改方法： insert、update、delete、select。
11  DISK和CONFIG FILE：
12     持久化配置信息。重启时，可以从磁盘快速加载回来。
```

#### Proxysql管理接口的三层配置文件关系

配置文件的信息都是以表的方式进行存储



### 3.3.2 ProxySQL配置语法

Bash | Copy

```
1 在不同层次间移动配置
2
3  LOAD xxxx  TO RUNTIME;
4  SAVE xxxx  TO DISK;
5
6  为了将配置持久化到磁盘或者应用到 runtime，在管理接口下有一系列管理命令来实现它们。
7  1. user相关配置
8  ## MEM 加载到runtime
9  LOAD MYSQL USERS TO RUNTIME;
10
11 ## runtime 保存至 MEM
12 SAVE MYSQL USERS TO MEMORY;
13
14 ## disk 加载到 MEM
15 LOAD MYSQL USERS FROM DISK;
16
17 ## MEM 到 disk
18 SAVE MYSQL USERS TO DISK;
19
20 ## CFG 到 MEM
21 LOAD MYSQL USERS FROM CONFIG
22 =====
23 2. server 相关配置
24 ## MEM 加载到runtime
25 LOAD MYSQL SERVERS TO RUNTIME;
26
27 ## runtime 保存至 MEM
28 SAVE MYSQL SERVERS TO MEMORY;
29
30 ## disk 加载到 MEM
31 LOAD MYSQL SERVERS FROM DISK;
32
33 ## MEM 到 disk
34 SAVE MYSQL SERVERS TO DISK;
35
36 ## CFG 到 MEM
37 LOAD MYSQL SERVERS FROM CONFIG
38 =====
39 3. mysql query rules配置
40 ## MEM 加载到runtime
41 LOAD MYSQL QUERY RULES TO RUNTIME;
42
43 ## runtime 保存至 MEM
44 SAVE MYSQL QUERY RULES TO MEMORY;
45
46 ## disk 加载到 MEM
47 LOAD MYSQL QUERY RULES FROM DISK;
48
49 ## MEM 到 disk
50 SAVE MYSQL QUERY RULES TO DISK;
```

```
51
52  ## CFG 到 MEM
53  LOAD MYSQL QUERY RULES FROM CONFIG
54
55  =====
56  4. MySQL variables配置
57  ## MEM 加载到runtime
58  LOAD MYSQL VARIABLES TO RUNTIME;
59
60  ## runtime 保存至 MEM
61  SAVE MYSQL VARIABLES TO MEMORY;
62
63  ## disk 加载到 MEM
64  LOAD MYSQL VARIABLES FROM DISK;
65
66  ## MEM 到 disk
67  SAVE MYSQL VARIABLES TO DISK;
68
69  ## CFG 到 MEM
70  LOAD MYSQL VARIABLES FROM CONFIG
71
72
73  总结:
74  日常配置其实大部分时间在MEM配置, 然后load到RUNTIME, 然后SAVE到Disk。cfg很少使用。
75  注意:
76  只有load到 runtime 状态时才会验证配置。
77  在保MEM或disk时, 都不会发生任何警告或错误。
78  当load到 runtime 时, 如果出现错误, 将恢复为之前保存得状态, 这时可以去检查错误日志。
```

## 4.ProxySQL应用案例-基于SQL的读写分离（使用正则进行匹配）

### 4.1 ProxySQL后端从节点设定为只读

因为ProxySQL根据read\_only参数进行主从的判断

```
1 从库设定read_only参数
2
3  set global read_only=1;
   set global super_read_only=1;
```

### 4.2 在mysql\_replication\_hostgroup表中, 配置读写组编号

Bash | Copy

```
1  insert into
2  mysql_replication_hostgroups
3  (writer_hostgroup, reader_hostgroup, comment)
4  values (10,20,'proxy');
5
6
7  load mysql servers to runtime;
8  save mysql servers to disk;
9
10 查看
11 db03 [(none)]>select * from mysql_replication_hostgroups ;
12 +-----+-----+-----+-----+
13 | writer_hostgroup | reader_hostgroup | check_type | comment |
14 +-----+-----+-----+-----+
15 | 10               | 20               | read_only  | proxy    |
16 +-----+-----+-----+-----+
17 ProxySQL会根据server 的read_only 的取值将服务器进行分组。
18 read_only=0 的server, master被分到编号为10的写组, read_only=1 的server, slave则被分到编号20的读组。所以需要将从库设置: se
```

### 4.3 ProxySQL后端主节点创建监控用户

Bash | Copy

```
1 create user monitor@'%' identified with mysql_native_password by '123';
2 grant replication client on *.* to monitor@'%';
```

### 4.4 ProxySQL修改variables表

Bash | Copy

```
1 mysql -uadmin -padmin -h127.0.0.1 -P6032
2
3
4 set mysql-monitor_username='monitor';
5 set mysql-monitor_password='123';
6
7
8 load mysql variables to runtime;
9 save mysql variables to disk;
```

### 4.5 添加主机到ProxySQL

Bash | Copy

```
1 mysql -uadmin -padmin -h127.0.0.1 -P6032
2
3 insert into mysql_servers(hostgroup_id,hostname,port) values (10,'10.0.0.51',3306);
4 insert into mysql_servers(hostgroup_id,hostname,port) values (20,'10.0.0.52',3306);
5 insert into mysql_servers(hostgroup_id,hostname,port) values (20,'10.0.0.53',3306);
6
7 load mysql servers to runtime;
8 save mysql servers to disk;
```

## 4.6 查看监控日志

Bash | Copy

```
1 mysql -uadmin -padmin -h127.0.0.1 -P6032
2
3 select * from mysql_server_connect_log;
4
5 | hostname | port | time_start_us | connect_success_time_us | connect_error |
6
7 | 10.0.0.53 | 3306 | 1621451953677107 | 412 | NULL |
8 | 10.0.0.51 | 3306 | 1621451954397147 | 1907 | NULL |
9 | 10.0.0.52 | 3306 | 1621451955117236 | 1679 | NULL |
10 | 10.0.0.51 | 3306 | 1621452013677560 | 2443 | NULL |
11 | 10.0.0.52 | 3306 | 1621452014472906 | 1779 | NULL |
12 | 10.0.0.53 | 3306 | 1621452015268154 | 398 | NULL |
13
14
15 select * from mysql_server_ping_log;
16 select * from mysql_server_read_only_log;
17 select * from mysql_server_replication_lag_log;
```

## 4.7 ProxySQL后端主节点创建应用用户

Bash | Copy

```
1 create user root@'%' identified with mysql_native_password by '123';
2 grant all on *.* to root@'%';
```

## 4.8 ProxySQL修改



Bash | Copy

```
1 insert into mysql_users(username,password,default_hostgroup) values('root','123',10);
2
3 load mysql users to runtime;
4 save mysql users to disk;
5
6 #早期版本，需要开启事务持续化。（现在的版本默认已经开启）
7 update mysql_users set transaction_persistent=1 where username='root';
8 load mysql users to runtime;
9 save mysql users to disk;
```

## 4.9 配置读写规则（正则书写规则，进行判断）

Bash | Copy

```
1 insert into mysql_query_rules(rule_id,active,match_pattern,destination_hostgroup,apply) values (1,1,'^select.*');
2 insert into mysql_query_rules(rule_id,active,match_pattern,destination_hostgroup,apply) values (2,1,'^select',2);
3
4 load mysql query rules to runtime;
5 save mysql query rules to disk;
6
7 注： select ... for update（会加锁，所以放在主库运行）规则的rule_id必须要小于普通的select规则的rule_id，ProxySQL是根据rule_id
```

## 4.10 测试读写分离

Bash | Copy

```
1 通过应用用户访问应用端口测试读写分离，proxysql根据读写规则进行读写分离
2 1.纯读的操作匹配到我们配置的读写规则rule_id=2，将操作分离到从节点
3 [root@db03 ~]# mysql -uroot -p123 -P 6033 -h 127.0.0.1 -e "select @@server_id;"
4 mysql: [Warning] Using a password on the command line interface can be insecure.
5 +-----+
6 | @@server_id |
7 +-----+
8 |          52 |
9 +-----+
10 [root@db03 ~]# mysql -uroot -p123 -P 6033 -h 127.0.0.1 -e "select @@server_id;"
11 mysql: [Warning] Using a password on the command line interface can be insecure.
12 +-----+
13 | @@server_id |
14 +-----+
15 |          53 |
16 +-----+
17 2.没有匹配到读写规则，会默认分离到default_hostgroup
18 mysql -uroot -p123 -P 6033 -h 127.0.0.1 -e "begin;select @@server_id;commit"
19 mysql: [Warning] Using a password on the command line interface can be insecure.
20 +-----+
21 | @@server_id |
22 +-----+
23 |          51 |
24 +-----+
```

## 4.11 查看读写分离路由的统计信息

执行的sql语句分离到那个节点的统计信息

Bash | Copy

```
1  mysql -uadmin -padmin -h127.0.0.1 -P6032
2
3  select * from stats_mysql_query_digest\G
4  ***** 1. row *****
5      hostgroup: 20
6      schemaname: information_schema
7      username: root
8      client_address:
9      digest: 0xDA65260DF35B8D13
10     digest_text: select @@server_id
11     count_star: 3
12     first_seen: 1621452912
13     last_seen: 1621452915
14     sum_time: 5897
15     min_time: 1074
16     max_time: 3725
17 sum_rows_affected: 0
18 sum_rows_sent: 3
19 ***** 2. row *****
20     hostgroup: 10
21     schemaname: information_schema
22     username: root
23     client_address:
24     digest: 0xDB3A841EF5443C35
25     digest_text: commit
26     count_star: 1
27     first_seen: 1621452905
28     last_seen: 1621452905
29     sum_time: 416
30     min_time: 416
31     max_time: 416
32 sum_rows_affected: 0
33 sum_rows_sent: 0
34
```

## 4.12 ProxySQL配置的步骤总结

Bash | Copy

- 1 0. 从节点设置只读
- 2 1. 加指定读组, 写组 hostgroup
- 3 2. 加监控用户
- 4 3. 加节点, 分配加入到读组, 写组中
- 5 4. 加读写分离的规则

## 5.ProxySQL应用扩展——花式路由规则(了解)

### 5.1 基于端口的路由

```
1  ## 修改ProxySQL监听SQL流量的端口号，监听多端口上。
2  set mysql-interfaces='0.0.0.0:6033;0.0.0.0:6034';
3  save mysql variables to disk;
4  ## 重启生效
5  systemctl restart proxysql
6
7  ## 设定路由规则
8  delete from mysql_query_rules;      # 为了测试，先清空已有规则
9
10 insert into mysql_query_rules(rule_id,active,proxy_port,destination_hostgroup,apply)
11 values(1,1,6033,10,1), (2,1,6034,20,1);
12
13 load mysql query rules to runtime;
14 save mysql query rules to disk;
15
16 说明：
17 除了基于端口进行分离，还可以基于监听地址(修改字段proxy_addr即可)，也可以基于客户端地址(修改字段client_addr字段即可)。
```

### 5.2 基于用户的路由

```
1  insert into mysql_users(username,password,default_hostgroup)
2  values('writer','123',10),('reader','123',20);
3  load mysql users to runtime;
4  save mysql users to disk;
5
6  delete from mysql_query_rules;      # 为了测试，先清空已有规则
7
8  insert into mysql_query_rules(rule_id,active,username,destination_hostgroup,apply)
9  values(1,1,'writer',10,1),(2,1,'reader',20,1);
10
11 load mysql query rules to runtime;
12 save mysql query rules to disk;
13
```

