DBAplus

金融行业运维实践-上海站

# 开源数据库云端灾备的应用实践

fuscott@yunify.com


# 开源数据库云端灾备的应用实践

fuscott@yunify.com

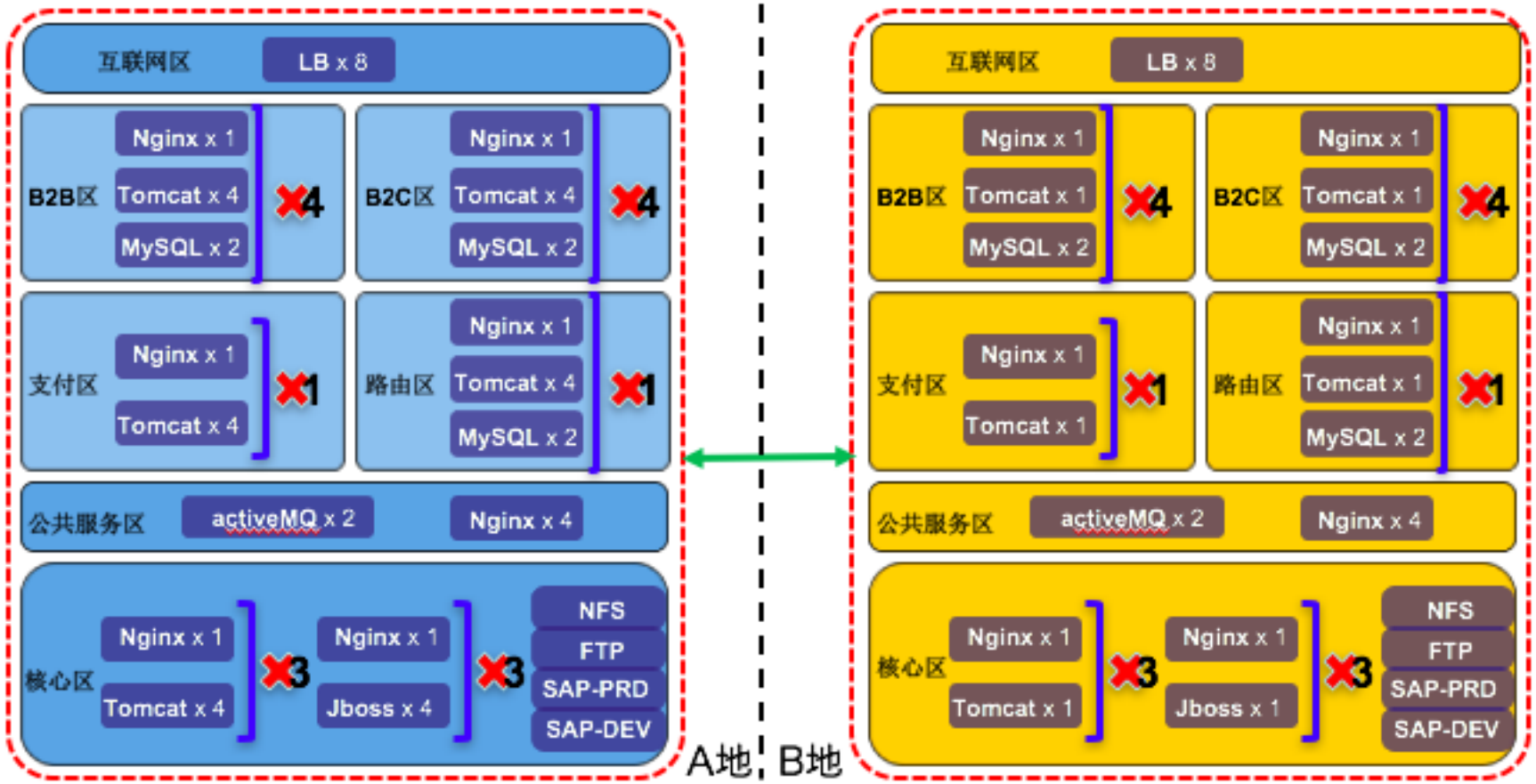
# 目录

- 背景情况
- 实践架构
- 技术展望

QINGCLOUD青云

# 背景情况

- 保险行业核心业务系统
- 面向互联网的金融创新业务
- 基于开源、分布式的业务系统架构



# 背景情况

- 以云计算平台做为IT基础设施
  小型机 / SAN存储 / VMware / 云计算平台
- 以分布式架构作为系统设计要求
  路由分片 / userid / redis
- 域名访问实现数据切换与调度
  内网域名改造 / 生产视图 / 灾备视图 / 演练视图
- 自动化运维工具
  统一发布系统 / CMDB / ITCS / 域名监控

# 实践架构

1、测试数据库部署位置

北京生产区、武汉灾备区

2、数据库规格配置

BJ Master Cluster : 2c/4G/100G  1主1从架构

WH Slave Cluster : 2c/4G/100G  1主1从架构

BJ Slave Cluster : 2c/4G/100G  1主1从架构

WH Master Cluster : 2c/4G/100G  1主1从架构

# 实践架构

## 步骤一：Master节点数据库提取GTID

SHOW MASTER STATUS;

## 步骤二：Master节点添加同步帐号与密码

```
GRANT REPLICATION SLAVE, REPLICATION CLIENT
ON *.*
TO 'repluser'@'%’
IDENTIFIED BY 'Zhu88jie';
FLUSH PRIVILEGES;
```

```
(root@10.130.12.214) [(none)]> show master status;
+------------------+----------+--------------+------------------+-------------------------------------------+
| File             | Position | Binlog_Do_DB | Binlog_Ignore_DB | Executed_Gtid_Set                         |
+------------------+----------+--------------+------------------+-------------------------------------------+
| mysql-bin.000002 |      191 |              |                  | 37d07c78-8f59-11e6-b376-ecf4bbec0432:1-13 |
+------------------+----------+--------------+------------------+-------------------------------------------+
1 row in set (0.01 sec)

(root@10.130.12.214) [(none)]> GRANT  REPLICATION  SLAVE,  REPLICATION  CLIENT
    -> ON  *.*
    -> TO  'repluser'@'%'
    -> IDENTIFIED  BY  'Zhu88jie';
Query OK, 0 rows affected (0.00 sec)

(root@10.130.12.214) [(none)]> FLUSH  PRIVILEGES;
Query OK, 0 rows affected (0.00 sec)
```

# 实践架构

步骤一：修改Master节点为GTID的同步方式

CHANGE MASTER TO

master_host='<master-cluster IP>',

master_user='repluser',

master_port=3306,（ VPC互联VPC为可配参数，VPC互联VBC为必配参数）

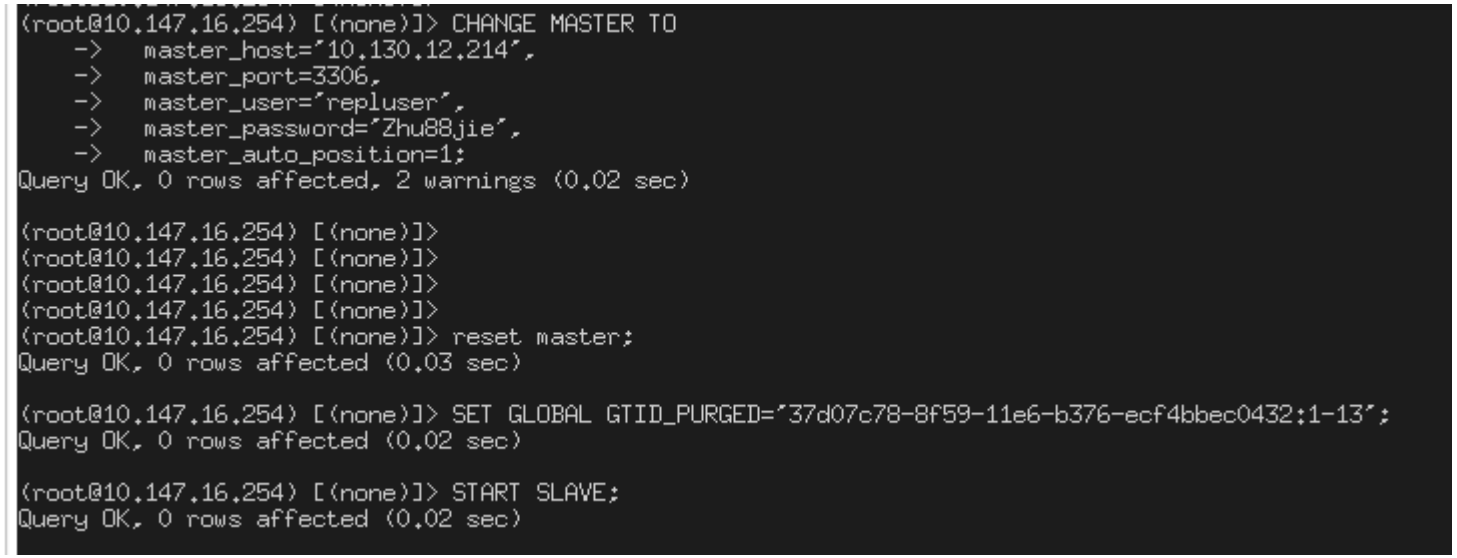
master_password='p12cHANgepwD',

master_auto_position=1;

RESET MASTER;

SET GLOBAL GTID_PURGED='<master-cluster Executed_Gtid_Set>';

START SLAVE;




QINGCLOUD青云

# 实践架构

步骤一：修改Slave节点以GTID的方式进行同步

STOP SLAVE; START SLAVE;

RESET MASTER;

SET GLOBAL GTID_PURGED='<master-cluster Executed_Gtid_Set>';

START SLAVE;

```
(root@10.147.16.254) [(none)]> reset master;
Query OK, 0 rows affected (0.03 sec)

(root@10.147.16.254) [(none)]> SET GLOBAL GTID_PURGED='37d07c78-8f59-11e6-b376-ecf4bbec0432:1-13';
Query OK, 0 rows affected (0.02 sec)

(root@10.147.16.254) [(none)]> START SLAVE;
Query OK, 0 rows affected (0.02 sec)
```

# 实践架构

## 步骤一：Master节点数据库提取GTID

SHOW MASTER STATUS;

## 步骤二：Master节点添加同步帐号与密码

```
 GRANT REPLICATION SLAVE, REPLICATION CLIENT
ON *.*
TO 'repluser'@'%’
IDENTIFIED BY 'Zhu88jie';
FLUSH PRIVILEGES;
```

```
(root@10.147.16.8) [(none)]>
(root@10.147.16.8) [(none)]> show master status \G;
*************************** 1. row ***************************
             File: mysql-bin.000001
         Position: 3364
     Binlog_Do_DB:
 Binlog_Ignore_DB:
Executed_Gtid_Set: fb6d2c58-903a-11e6-854c-246e960d0bca:1-13
1 row in set (0.01 sec)

ERROR:
No query specified

(root@10.147.16.8) [(none)]>
(root@10.147.16.8) [(none)]>
(root@10.147.16.8) [(none)]>
(root@10.147.16.8) [(none)]>
(root@10.147.16.8) [(none)]>
(root@10.147.16.8) [(none)]> GRANT  REPLICATION  SLAVE,  REPLICATION  CLIENT
    -> ON  *.*
    -> TO  'repluser'@'%'
    -> IDENTIFIED  BY  'Zhu88jie';
Query OK, 0 rows affected (0.02 sec)

(root@10.147.16.8) [(none)]> flush privileges ;
Query OK, 0 rows affected (0.02 sec)
```

QINGCLOUD青云

# 实践架构

步骤一：修改Master节点为GTID的同步方式

CHANGE MASTER TO

master_host='<master-cluster IP>',

master_user='repluser',

master_port=3306,（VPC互联VPC为可配参数，VPC互联VBC为必配参数）

master_password='p12cHANgepwD',

master_auto_position=1;

RESET MASTER;

SET GLOBAL GTID_PURGED='<master-cluster Executed_Gtid_Set>';

START SLAVE;

```
(root@100.127.0.9) [(none)]>
(root@100.127.0.9) [(none)]>
(root@100.127.0.9) [(none)]> CHANGE MASTER TO
    ->   master_host='10.147.16.8',
    ->   master_port=3306,
    ->   master_user='repluser',
    ->   master_password='Zhu88jie',
    ->   master_auto_position=1;
Query OK, 0 rows affected, 2 warnings (0.02 sec)

(root@100.127.0.9) [(none)]> reset master ;
Query OK, 0 rows affected (0.01 sec)

(root@100.127.0.9) [(none)]>
(root@100.127.0.9) [(none)]>
(root@100.127.0.9) [(none)]> SET GLOBAL GTID_PURGED='fb6d2c58-903a-11e6-854c-246e960d0bca:1-13
';
Query OK, 0 rows affected (0.02 sec)

(root@100.127.0.9) [(none)]>
(root@100.127.0.9) [(none)]> start slave ;
Query OK, 0 rows affected (0.00 sec)
```

# 实践架构

步骤一：修改Slave节点以GTID的方式进行同步

STOP SLAVE; START SLAVE;

RESET MASTER;

SET GLOBAL GTID_PURGED='<master-cluster Executed_Gtid_Set>';

START SLAVE;

```
(root@100.127.0.8) [(none)]> stop slave ;
Query OK, 0 rows affected (0.01 sec)

(root@100.127.0.8) [(none)]> start slave ;
Query OK, 0 rows affected (0.00 sec)

(root@100.127.0.8) [(none)]> reset master ;
Query OK, 0 rows affected (0.01 sec)

(root@100.127.0.8) [(none)]> SET GLOBAL GTID_PURGED='fb6d2c58-903a-11e6-854c-246e960d0bca:1-13
';
Query OK, 0 rows affected (0.01 sec)

(root@100.127.0.8) [(none)]> start slave
    -> ;
Query OK, 0 rows affected, 1 warning (0.00 sec)
```

# 实践架构

数据库测试脚本（1000万）

```
create database testdb ;

create table testdb.testdata(id int(11), content varchar(255), date1 TIMESTAMP(6));

use testdb;

DELIMITER $$

create procedure pro11()

-> begin

-> declare i int default 0;

-> repeat

-> insert into testdb.testdata values(i,concat('adasfsdfFJDSFJISDFJFSFFFF',i),now());

-> set i=i+1;

-> until i>=10000000

-> end repeat;

-> end$$

DELIMITER ;

select now();

call pro11();
```

# 实践架构

应用组件环境配置

1 * LB（4C/8G） 1 * nginx（4C/8G） 2 * tomcat（4C/8G）

**应用测试场景说明（Loadrunner压力生成）：**

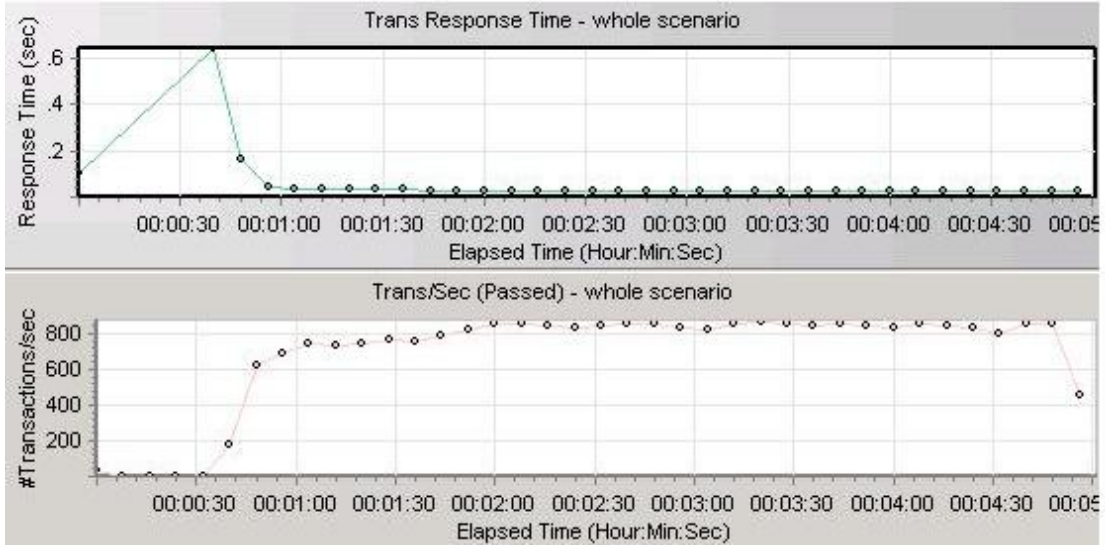
测试1：一定数量的交易（5分钟20万笔交易）

测试2：一定时间的测试（30分钟连续压测）

测试3：灾难模拟测试

# 实践架构

**时间：5分钟  并发用户：50并发**
**平均响应时间：0.033秒 TPS：673.651笔/秒  成功交易数：208215笔**
**VM CPU使用率：app01 75% app02 75% web01 10%**

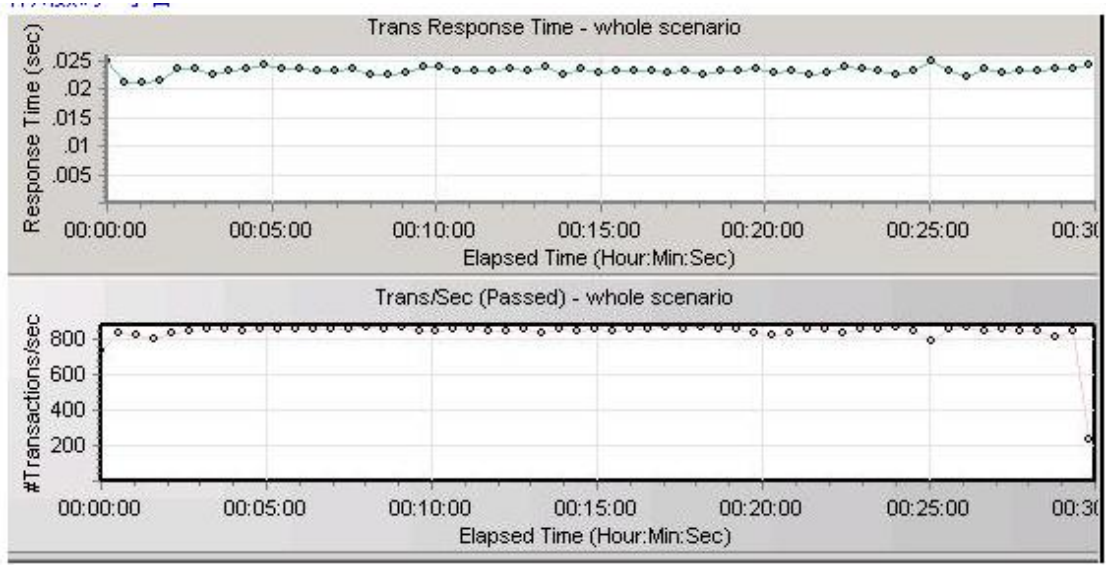



| 序号 | 统计时间 | 环境 | 表s_trade | 表s_verifylog | 延时/说明 |
|---|---|---|---|---|---|
| 测试开始 | 2016/4/14 20:33 | 北京主库 | 86515 | 86826 | |
| | 2016/4/14 20:33 | 武汉主库 | 86515 | 86826 | |
| 测试结束 | 2016/4/14 20:38 | 北京主库 | 294730 | 295041 | 294730-86515=208215 |
| | 2016/4/14 20:38 | 武汉主库 | 294730 | 295041 | ***295041-86826=208215*** |

QINGCLOUD青云

# 实践架构

**时间：30分钟  并发用户：50并发**
**平均响应时间：0.023秒 TPS：826.502笔/秒  成功交易数：1527140笔**
**VM CPU使用率：app01 75% app02 75% web01 10%**
**北京主库带宽（进28Mbps出56Mbps）CPU28%　　武汉主库带宽（进出均为18Mbps）CPU10%**




| 序号 | 统计时间 | 环境 | 表s_trade | 表s_verifylog | 延时/说明 |
|---|---|---|---|---|---|
| 时间点1 | 2016/4/14 20:49 | 北京主库 | 425081 | 425508 | |
| | 2016/4/14 20:49 | 武汉主库 | 425847 | 426293 | 北京和武汉无延迟 |
| 时间点2 | 2016/4/14 20:55 | 北京主库 | 689447 | 689955 | 3006/848=3.5s |
| | 2016/4/14 20:55 | 武汉主库 | 686441 | 686916 | 北京和武汉延迟3.5s |
| 时间点3 | 2016/4/14 21:00 | 北京主库 | 982695 | 983273 | 6,291/848=7.42s |
| | 2016/4/14 21:00 | 武汉主库 | 976404 | 976973 | 武汉延迟7.42s |
| 时间点4 | 2016/4/14 21:05 | 北京主库 | 1240943 | 1241543 | 11089/848=13.07s |
| | 2016/4/14 21:05 | 武汉主库 | 1229854 | 1230544 | 武汉延迟13.07s |
| 时间点5 | 2016/4/14 21:10 | 北京主库 | 1476944 | 1477670 | 9045/848=10.67s |
| | 2016/4/14 21:10 | 武汉主库 | 1467899 | 1468577 | 武汉延迟10.67s |
| 时间点6 | 2016/4/14 21:16 | 北京主库 | 1776173 | 1776868 | 7234/848=8.53s |
| | 2016/4/14 21:16 | 武汉主库 | 1768939 | 1769632 | 武汉延迟8.53s |
| 时间点7 | 2016/4/14 21:17 | 北京主库 | 1821870 | 1822181 | 1821870-294730=1,527,140 |
| | 2016/4/14 21:17 | 武汉主库 | 1821870 | 1822181 | 与压测显示的成功笔数一致 |

# 实践架构

1、应用故障：不影响数据的同步，数据可以保持一致，延时规律与测试项目2一致，业务恢复取决于应用恢复/切换时间。

2、数据库故障：本次测试为关闭数据库组件服务，但是由于binlog仍然存在并可以同步，两边数据仍能保持一致；若宕机导致无法访问binlog，将会丢失部分数据（实验2中延时范围内的数据），是否能完全恢复取决于北京主库binlog日志是否可以还原或者应用日志是否可以提取恢复。

3、北京基础环境及网络等发生灾难：

22:31:37 中断GRE隧道

22:32:04 停止nginx 、app、LB等

灾难发生后，两边数据不一定一致，取决于灾难如何发生，以及发生后具体情况而定数据是否能完全恢复。丢失量约为：实验2中延时时间范围内的数据

| 分项 | 统计时间 | 环境 | 表s_trade | 表s_verifylog |
|---|---|---|---|---|
| 1、 应用故障<br>ngxin，tomcat均宕机 | 故障前 | 北京主库 | 1821870 | 1822181 |
| | 故障前 | 武汉主库 | 1821870 | 1822181 |
| | 故障后 | 北京主库 | 2084256 | 2084567 |
| | 故障后 | 武汉主库 | 2084256 | 2084567 |
| 2、 数据库故障<br>数据库宕机 | 故障前 | 北京主库 | 2091602 | 2091913 |
| | 故障前 | 武汉主库 | 2091602 | 2091913 |
| | 故障后 | 北京主库 | 2401356 | 2401667 |
| | 故障后 | 武汉主库 | 2401356 | 2401667 |
| 3、 基础资源及网络所有<br>发生灾难 | 故障前 | 北京主库 | 2401356 | 2401667 |
| | 故障前 | 武汉主库 | 2401356 | 2401667 |
| | 故障后 | 北京主库 | 2735442 | 2735753 |
| | 故障后 | 武汉主库 | 2712419 | 2712734 |

# 实践架构

测试场景-1结论

事务数与压力成功事务数一致 ，北京和武汉基本同时完成 。

测试场景-2结论

该测试最大延迟在15s以内，当交易数据不高时，基本无延迟。

测试场景-3结论

1、应用故障：不影响数据的同步，数据可以保持一致，延时规律与测试项目2一致，业务恢复取决于应用恢复/切换时间。

2、数据库故障：本次测试为关闭数据库组件服务，但是由于binlog仍然存在并可以同步，两边数据仍能保持一致；若宕机导致无法访问binlog，将会丢失部分数据（实验2中延时范围内的数据），是否能完全恢复取决于北京主库binlog日志是否可以还原或者应用日志是否可以提取恢复。

3、北京基础环境及网络等发生灾难：

灾难发生后，两边数据不一定一致，取决于灾难如何发生，以及发生后具体情况而定数据是否能完全恢复。丢失量约为：实验2中延时时间范围内的数据

# 技术展望

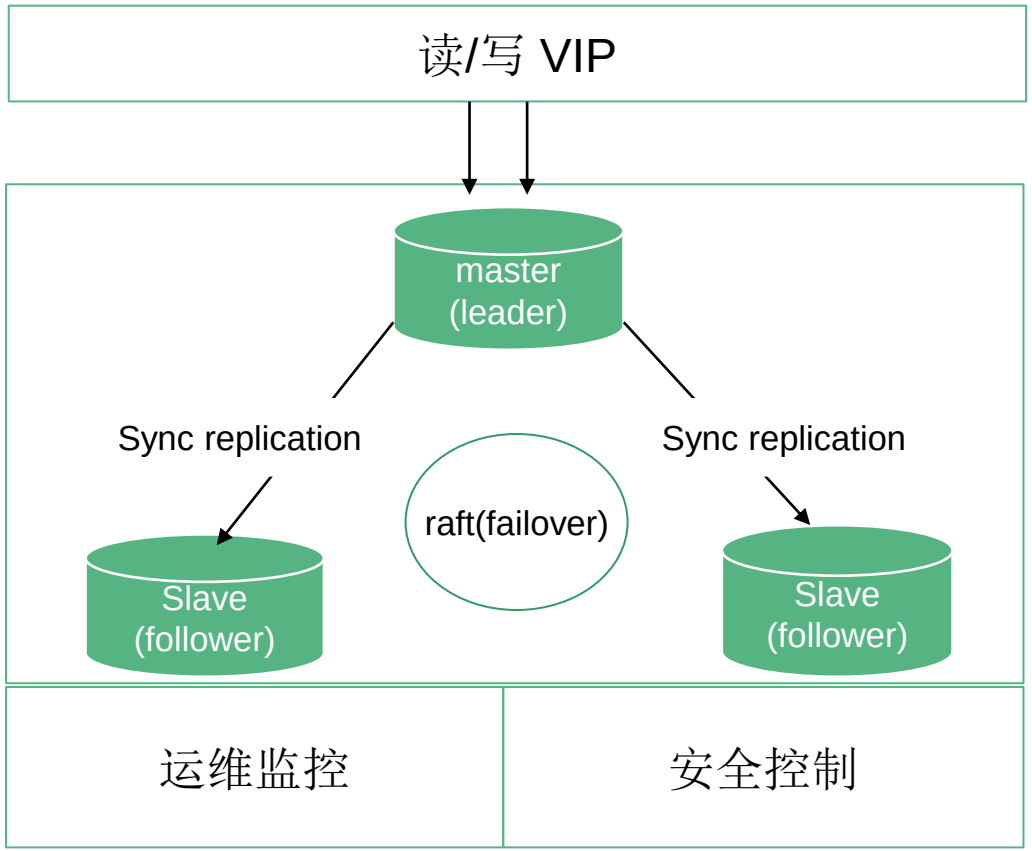
- 分片规则调整需要相关业务系统中间件重启
- 业务系统对数据库的压力无法实现自动调度
- 单库容量受限

# 技术展望

分布式数据库对业务系统支撑能力的体现

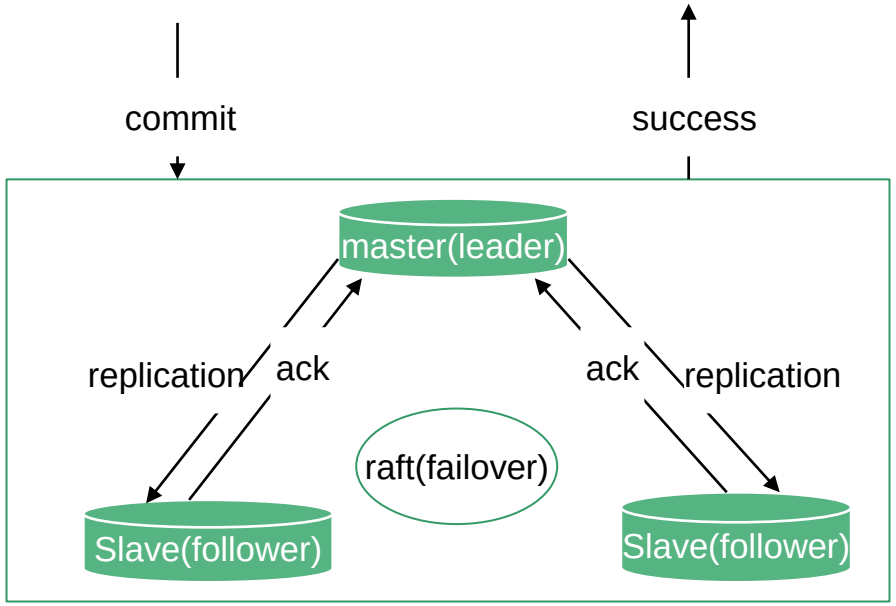
- 高性能 – SQL优化 / 连接池优化 / 网络与存储性能优化
- 高可靠 – 基于类似RAFT+GTID的高可用架构 / 同城异地容灾架构
- 可扩展 – 自动分库分表 / 自动存储扩容

# MySQL PLUS 数据库架构



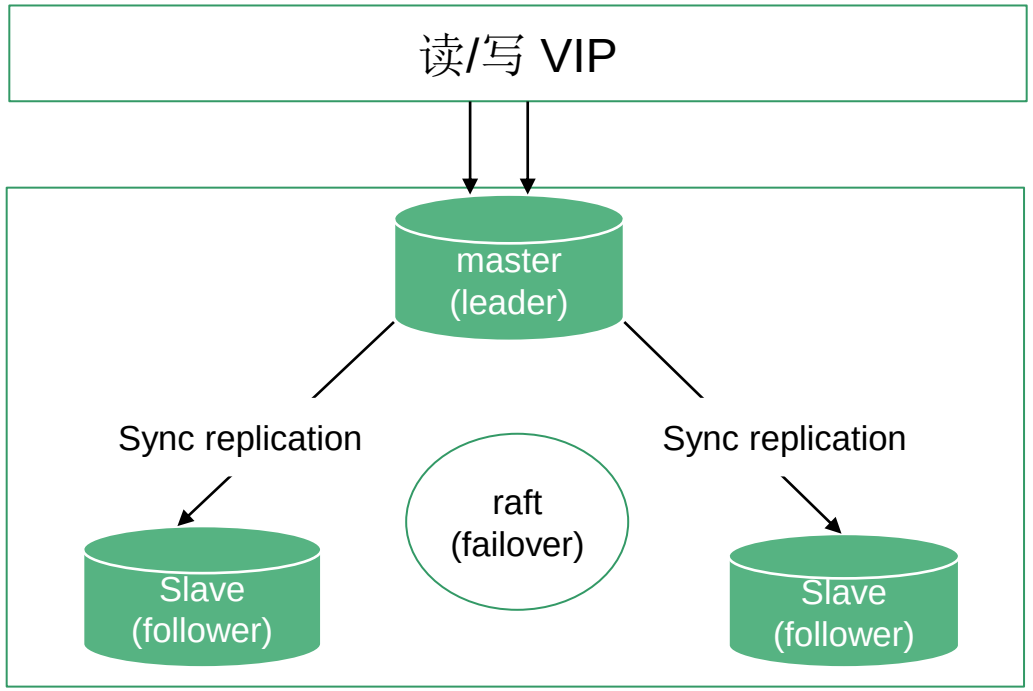
# 数据一致性--同步复制

- 在master上提交事务并写入binlog后，需要收到绝大多数slave节点已完成该事务的确认
- 如果master没有收到确认，将一直等待，直到成功



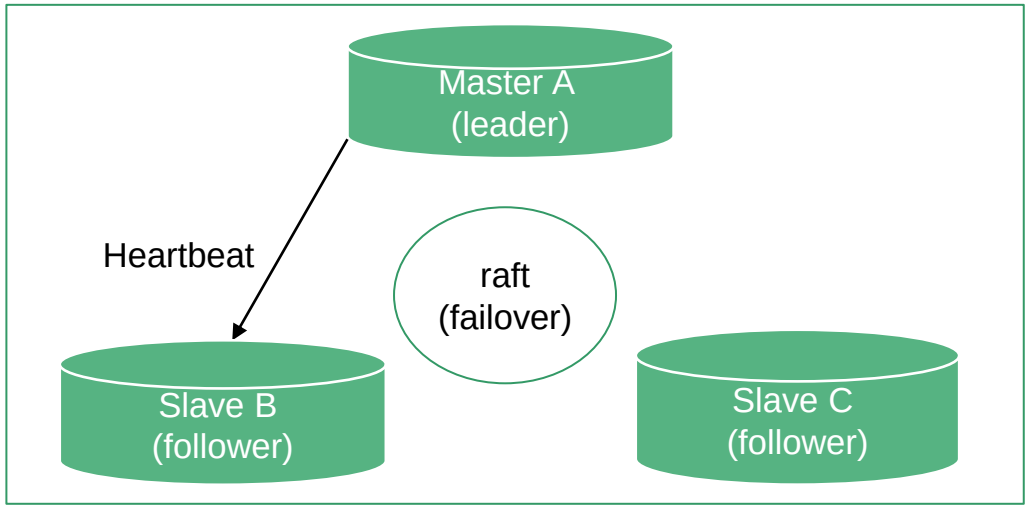
# 业务高可用--服务高可用

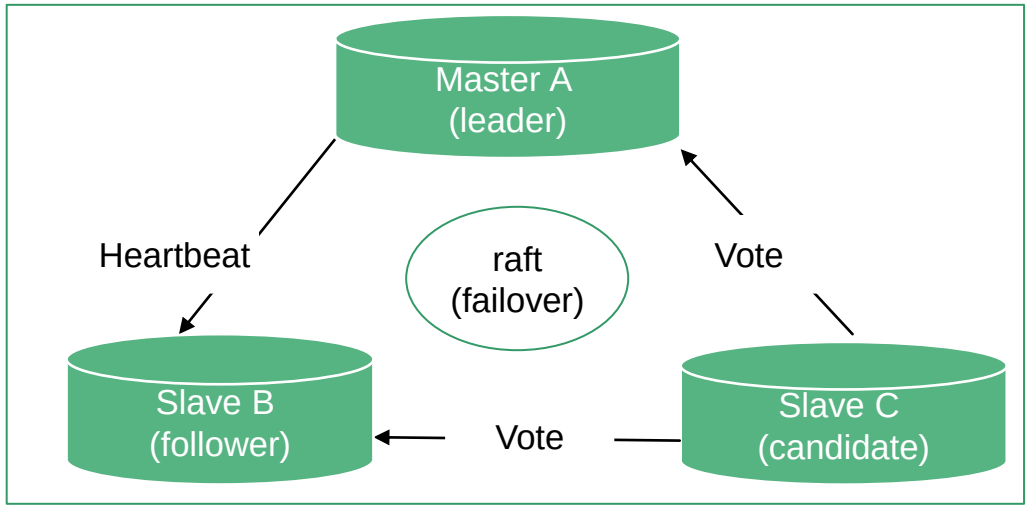
- 一主多从
- Raft 协议秒级切换
- 高可用 VIP



# 服务高可用--Raft选主过程

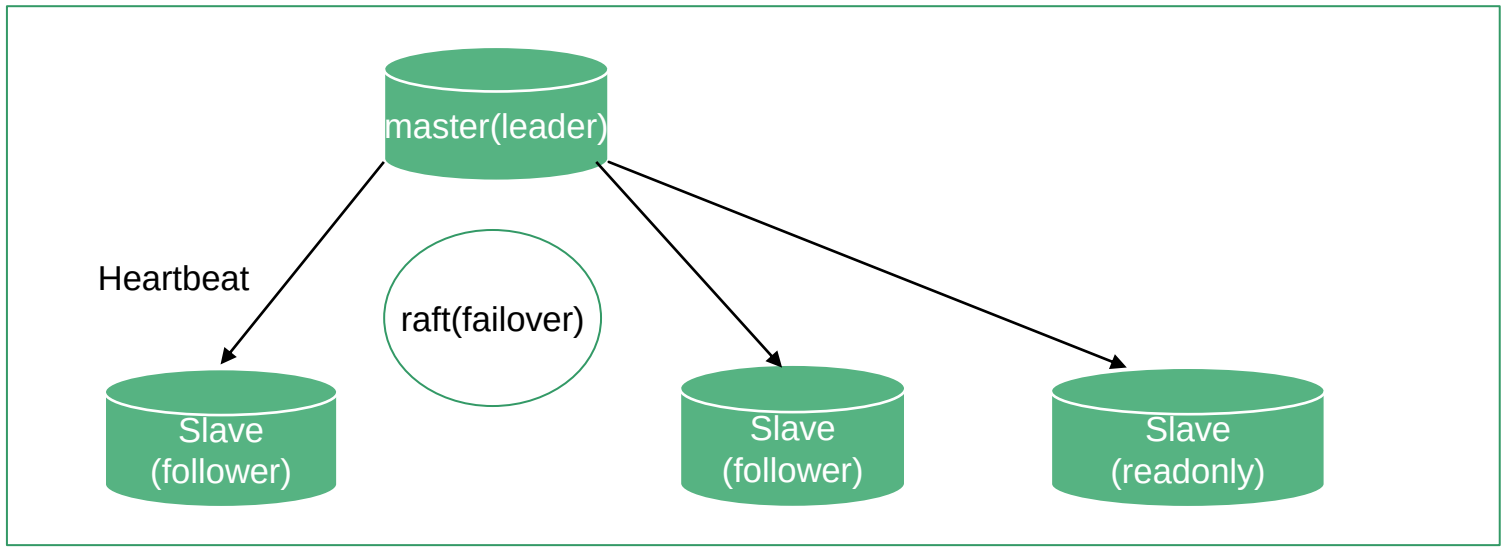
- 利用raft协议,结合GTID来选主,选主之后还是通过MySQL来做复制
- Leader: 定期发送心跳
- Follower:接收同步消息
- Candidate: 候选人,发起新一轮选举



# 服务高可用--只读节点

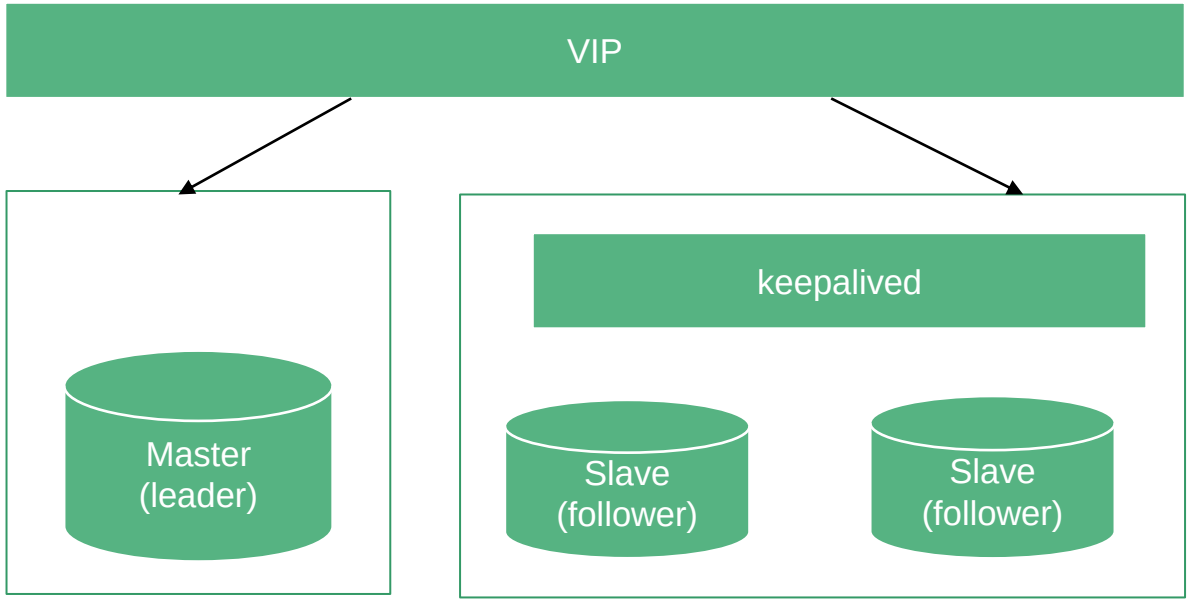
- 不参与选主,不主动投票
- 其他成员不认可它的投票权
- 新主产生时重置主从关系



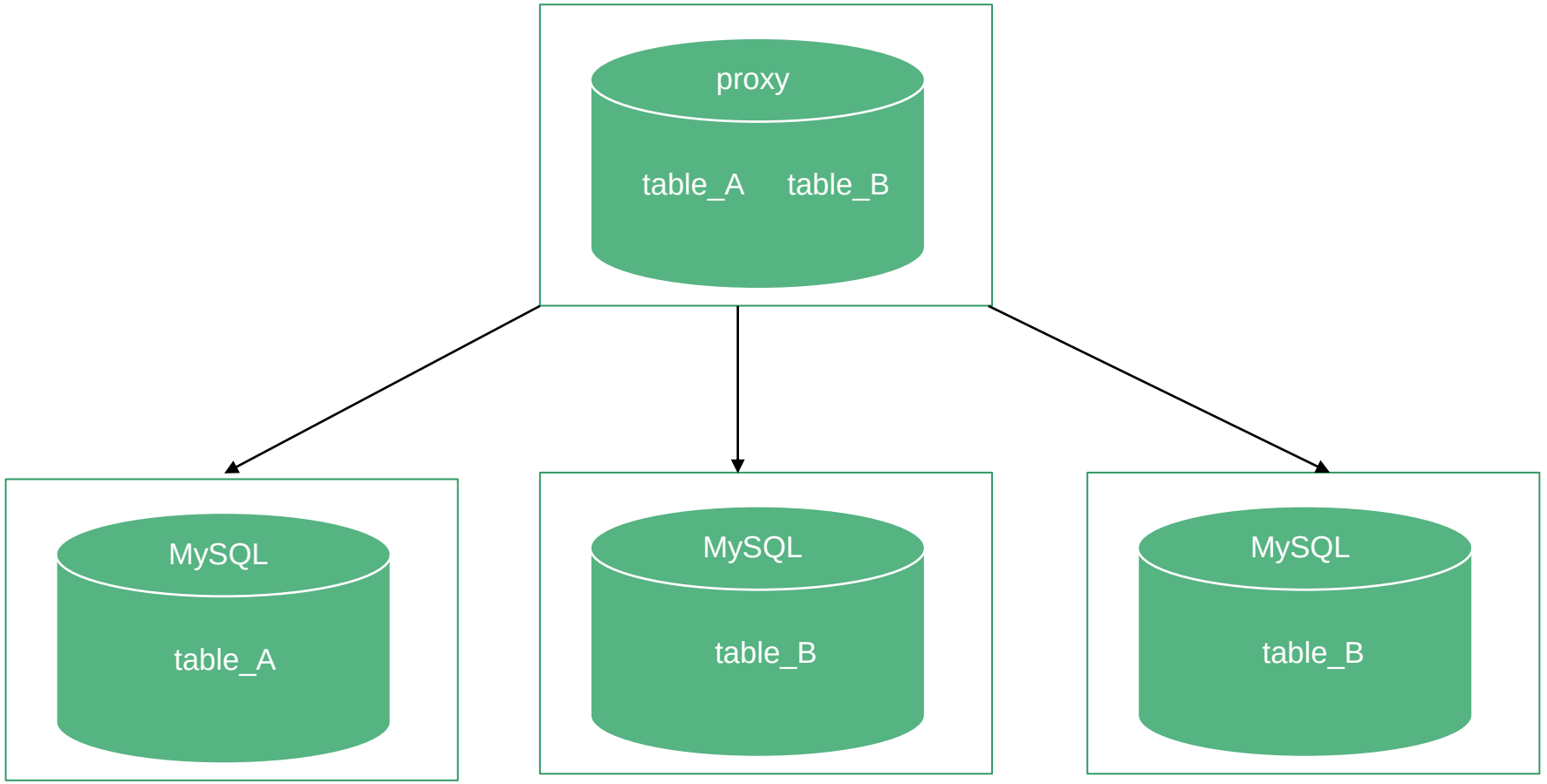

QINGCLOUD青云

# 服务高可用--读写VIP

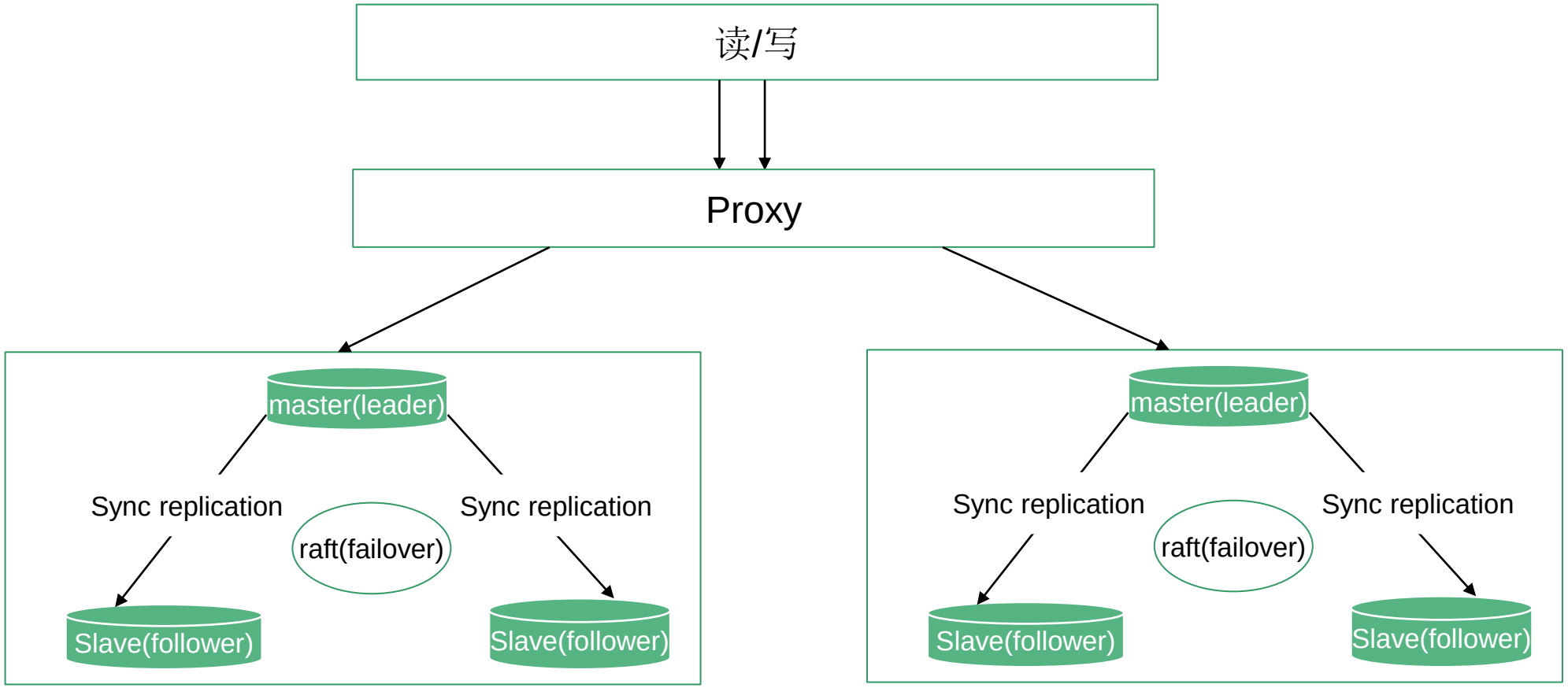
- 读写VIP不随角色变化
- 动态修改VIP配置,摘掉故障节点



# 海量存储--分布式数据库



# 海量存储--分布式数据库架构




QINGCLOUD青云


# Thank you.

fuscott@yunify.com

限时优惠，扫码抢票


# APAC OTN TOUR 2017

The APAC OTN Tour 2017 will be running from November 20th until December 9th visiting 4 countries/7 Cities in the Asia Pacific Region. Bellow you can find more information regarding the events that are part of this year tour:

**Dates:**

- Wellington, NZ : November 20th
- Auckland, NZ : November 22nd
- Sydney, Australia: November 24th
- Melbourne, Australia: November 27th
- Perth, Australia: November 29th
- Shanghai, China: December 3rd
- Hyderabad, India: December 8 and 9

12月3日 D+ Day 欢迎来撩：
讲师、茶歇、场地、赞助，统统可以。

预计150人规模。
微信来撩：boypoo

QINGCLOUD青云