# 数据库迁移方案

## 一、引言

### 1.1编写目的

为数据库迁移做详细规划

### 1.2项目背景

为了实现数据库的高可用，负载均衡和读写分离以提高数据库的整体性能。先由一主多从的的架构迁移到Atlas+MHA+MySQL replication上

### 1.3目标读者

数据库管理员

## ****二、迁移步骤****

### 2.1系统部署

**（atlas+mha+mysql Master+mysql slave）**

|  |  |  |  |
| --- | --- | --- | --- |
| IP地址 | 内网IP | MySQL版本 | 角色 |
| 120.26.103.172 | 10.252.86.134 | 5.6.23 | 目前生产主库 |
| 120.26.137.247 | 10.51.23.196 | 5.6.23 | 新环境（主库） |
| 120.26.71.71 | 10.252.124.43 | 5.6.23 | 新环境（主备） |
| 121.40.166.38 | 10.168.6.6 | 5.6.23 | 新环境（从库） |
| 120.26.44.153 | 10.175.204.158 | 5.6.23 | 新环境（数据仓库） |
| 120.26.127.218 | 10.252.82.111 | Atlas-2.2.1  mha4mysql-manager-0.56 | 新环境（Atlas，MHA） |
| 121.40.17.62 | 10.251.233.151 | Atlas-2.2.1  mha4mysql-manager-0.56 | 新环境（Atlas，MHA）镜像 |

**参照svn中数据库新架构配置文档配置**

### 2.2 数据移植

迁移前准备

### 2.2.1原库准备

在生产主库(120.26.103.172)上导出整库的数据

Mysqldump --all-databases --flush-logs --single-transaction --master-data=2 –events –routines>/mydata/all\_databases.sql

将innodb\_buffer\_pool进行备份

set global innodb\_buffer\_pool\_dump\_now=on;

会在数据目录下生成 ib\_buffer\_pool文件

将备份数据copy到新环境主库（10.51.23.196）上

scp /mydata/all\_databases.sql 10.51.23.196:/mydata/

scp /mydata/mysql/data/ib\_buffer\_pool **10.51.23.196**:/mydata/mysql/data/

## 2.2.2新库准备

a.在新环境主库（120.26.137.247）上导入整库的数据

*mysql–h127.0.0.1 –P3316 mysql </mydata/all\_databases.sql*

b.在新环境主库（120.26.137.247）上配置主从复制

配置生产主库(120.26.103.172) 和新环境主库（120.26.137.247）的主从关系

根据生产主库(120.26.103.172)上现有的复制账户和密码获取master\_user 和master\_password；

more /mydata/all\_databases.sql 获取 master\_log\_file和master\_log\_pos;

*Change master to master\_host=’* 120.26.103.172*’,master\_port=3316,master\_user=’XX’,master\_password=’XX’,master\_log\_file=’XX’,master\_log\_pos=XX;*

*Start slave;*

*Show slave status\G;*

查看复制状态 I/O\_threads 和SQL\_threads 都为YES即可

Chown -R mysql:mysql /mydata/mysql/data/ib\_buffer\_pool

mysql> set global innodb\_buffer\_pool\_load\_now=1;

mysql>show status like ‘innodb%’;

开始迁移

### 2.3 切换生产库

a.原生产库上设置锁表只读：flush tables with read lock; 或者直接停掉应用

b.新环境主库的POS点与原主库同步不变时

c.检测数据一致性，如果一致，新主flush logs;

d.将应用的数据库连接改为新环境连接

新主与新从之间一致性检查：

pt-table-checksum –nocheck-replication-filters –database=test –replicate=test.checksums –host=10.51.23.196 --port=3316 --user= --password= --no-check-binlog-format

新主与原主之间一致性检查：

由于原库设置为只读，所以只能使用脚本来检测

Compare\_diffs.sh

## 三、新系统上线后的工作

将新库的event设置为enable

Alter event ENT\_ACCOUNT\_CHANGE enable；

Alter event ENT\_ACCOUNT\_BANK\_UPDATE enable；

Alter event ENT\_PAYMENT\_UPDATE enable；

## ****四、回退方案****

如果切换到新架构后，出现不可预测的问题时，需要回滚

1. 如果数据没问题，架构出现没预测到的问题，则全备新主,copy到原主，并导入

新主：Flush tables with read lock;

Mysqldump --all-databases --flush-logs --single-transaction --master-data=2 –events –routines>/mydata/new\_all\_databases.sql

cp /mydata/new\_all\_databases.sql 10.252.86.134:/mydata/

原主：mysql account</mydata/new\_all\_databases.sql

应用指向原主

1. 如果数据有问题，则停止读写分离；配合研发找出问题数据并修复；

在atlas的管理端口执行

mysql -h10.252.82.111 -P3315 -uadmin -p'admin'

select \* from backends；

add slave 10.51.23.196:3316；

remove backend $backend\_id；($backend\_id 为ro的从库的id)