数据库管理

|  |  |  |  |
| --- | --- | --- | --- |
| 拟制： | **梁锋** |  |  |
| 审核: |  |  |  |
| 批准: |  |  |  |
| 日期： | 2017年06月06日 |  |  |

修订记录

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 序号 | 完成日期 | 更改内容描述 | 拟制修订 | 审核 | 批准 |
| 1 | 年月日 |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |

目录

[第1章. 目的 4](#_Toc488055964)

[第2章. 适用范围 5](#_Toc488055965)

[第3章. 点金乐购数据库备份机制 6](#_Toc488055966)

[3.1 全备机制 6](#_Toc488055967)

[3.2 增备机制 6](#_Toc488055968)

[第4章. 点金乐购数据库主从机制 6](#_Toc488055969)

[4.1 主从同步监控 6](#_Toc488055970)

[4.2 主从状态手动检查 7](#_Toc488055971)

[4.2.1 登录从库服务器 7](#_Toc488055972)

[4.2.2 检查主从状态（正常） 7](#_Toc488055973)

[4.2.3 检查主从状态（异常） 8](#_Toc488055974)

[4.2.4 主从修复（主从同步异常时操作） 8](#_Toc488055975)

[4.2.5 注意事项 10](#_Toc488055976)

[第5章. 数据库服务器巡检 11](#_Toc488055977)

[5.1 数据库进程状态 11](#_Toc488055978)

[5.2 数据库备份文件检查 11](#_Toc488055979)

[5.2.1 全量备份文件检查 11](#_Toc488055980)

[5.2.2 增量备份是否正常，历史备份文件数量是否正常。 12](#_Toc488055981)

[5.3 数据库日志文件检查 12](#_Toc488055982)

[5.4 数据库服务器容量预估 12](#_Toc488055983)

[第6章. 数据库性能优化（提供的MySQL已经过优化） 13](#_Toc488055984)

[6.1 最大连接数 13](#_Toc488055985)

[6.2 innodb使用内存 13](#_Toc488055986)

[6.3 innodb\_log\_buffer\_size 13](#_Toc488055987)

[6.4 innodb\_thread\_concurrency 13](#_Toc488055988)

[6.5 myisam 13](#_Toc488055989)

[6.6 慢查询日志 13](#_Toc488055990)

[第7章. 数据库性能巡检 14](#_Toc488055991)

[7.1 最大连接数 14](#_Toc488055992)

[7.2 命中率 14](#_Toc488055993)

[7.3 临时表 14](#_Toc488055994)

[7.4 表打开数 15](#_Toc488055995)

[7.5 进程使用 15](#_Toc488055996)

[7.6 文件打开数 15](#_Toc488055997)

[第8章. 附录(无) 16](#_Toc488055998)

[第9章. 附则(无) 17](#_Toc488055999)

# 目的

为了让运维人员尽快熟悉数据库常见的基本管理工作，特制定此文档。

# 适用范围

此文档适用于具备Linux基础和MySQL基础的技术人员。

# 点金乐购数据库备份机制

## 全备机制

每天结算前和开盘前进行两次全量备份。全量备份数据保留两份。一份在数据库服务器本地，一份在交易所专用备份服务器。实现数据备份冗余。

备份文件的具体存放地址根据交易所服务器硬盘情况规划，在交易所上线后提交给交易所。

## 增备机制

每小时执行一次增备操作，增备文件存放在交易所专用备份服务器。

备份文件的具体存放地址根据交易所服务器硬盘情况规划，在交易所上线后提交给交易所。

# 点金乐购数据库主从机制

点金乐购产品使用mysql数据库，启用mysql主从同步实现数据同步和读写分离，提升数据库性能并能保证数据实时备份。因此主从同步状态要保证正常运行。

## 主从同步监控

zabbix监控实现，此处略过

## 主从状态手动检查

### 登录从库服务器

### 检查主从状态（正常）



### 检查主从状态（异常）



### 主从修复（主从同步异常时操作）

#### 方法一：忽略错误后，继续同步

mysql>stop slave;

mysql>set global sql\_slave\_skip\_counter =1; #表示跳过一步错误，后面的数字可变

mysql>start slave;

mysql> show slave status\G

Slave\_IO\_Running: Yes

Slave\_SQL\_Running: Yes

#### 方式二：重新做主从，完全同步

1.先进入主库，进行锁表，防止数据写入

mysql> flush tables with read lock;

2.数据拷贝 (**在从库操作**)

备份从库server-uuid

#cp /data/mysql/auto.cnf /tmp/auto.cnf

#cd /data/mysql

#rm -rf \*

把主库数据拷贝到从库

#scp -R /data/mysql/\* root@从库IP:/data/mysql/

还原从库server-uuid

#cp /tmp/auto.cnf /data/mysql/

3.查看master 状态

mysql> show master status;

+-------------------+----------+--------------+-------------------------------+

| File | Position | Binlog\_Do\_DB | Binlog\_Ignore\_DB |

+-------------------+----------+--------------+-------------------------------+

| mysqld-bin.000001 | 3260 | | mysql,test,information\_schema |

+-------------------+----------+--------------+-------------------------------+

1 row in set (0.00 sec)

4.停止从库的状态

mysql> stop slave;

5.设置从库同步，注意该处的同步点，就是主库show master status信息里的| File| Position两项

mysql> change master to

master\_host = '192.168.128.100',

master\_user = user,

master\_port=xxxx,

master\_password='123456',

master\_log\_file = 'mysqld-bin.000001',

master\_log\_pos=3260;

6.重新开启主从同步

mysql> start slave;

7.查看同步状态

mysql> show slave status\G

Slave\_IO\_Running: Yes

Slave\_SQL\_Running: Yes

8.断开主库连接，释放只读锁。

### 注意事项

1，需要根据造成主从断开的错误原因来判断应该选择哪一种修复方式，忽略错误继续同步操作可能或造成主从数据不一致，只有在类似表结构冲突导致的主从断开的情况下才可以选择该模式。为确保数据一致性优先选择重新做主从。

2，重做主从操作必须在休市状态下操作，锁表状态下数据库无法访问，系统不可用。

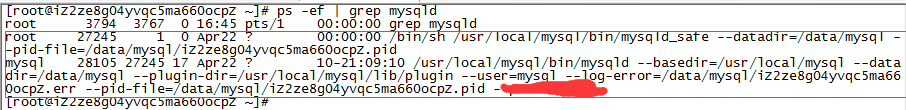
# 数据库服务器巡检

## 数据库主从状态

参照第四章内容。

## 数据库进程状态

确保以下两个数据库进程存在。



## 数据库备份文件检查

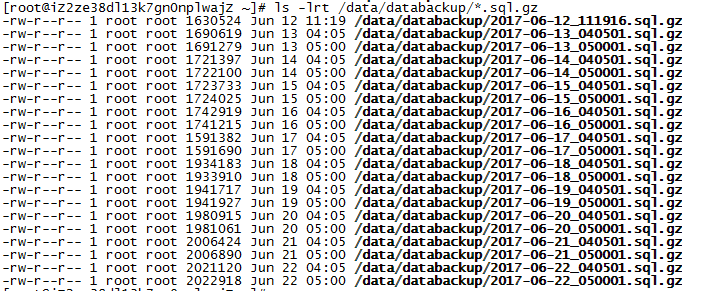
首先确认备份任务执行时间及备份文件存放目录，本文档不做备份路径说明。

ls -lrt /data/database\_backup/\*.sql.gz

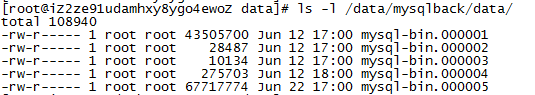
确认最新的sql.gz文件是否正常

### 全量备份文件检查

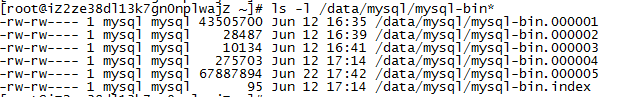
全量备份是否正常，历史备份文件数量是否正常；备份文件大小是否正常。



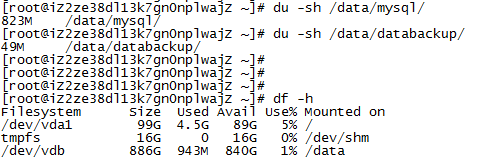
### 增量备份是否正常，历史备份文件数量是否正常。



## 数据库日志文件检查



## 数据库服务器容量预估



当数据库磁盘空间占用过大时需要控制备份文件数量和日志文件，如果备份文件和日志不希望被删除，数据库磁盘空间需要扩容。

# 数据库性能优化（提供的MySQL已经过优化）

## 最大连接数

设置最大连接数为2000，保证数据库不会因为连接数过大导致不可用。

## innodb使用内存

提升Innodb内存值为总内存的3/4。

## innodb\_log\_buffer\_size

减少IO资源消耗，提升数据读写性能。

## innodb\_thread\_concurrency

对mysql的线程并发数不做限制。

## myisam

myisam参数调整，保证myisam性能。

## 慢查询日志

开启慢查询日志，执行时间超过两秒的查询语句被记录下来，便于应用调优。

# 数据库性能巡检

## 最大连接数

在数据库连接工具或登录数据库服务器后台执行命令：

show global status like 'Max\_used\_connections';

showvariables like 'max\_connections';

计算：

Max\_used\_connections / max\_connections \* 100% <85%

## 命中率

在数据库连接工具或登录数据库服务器后台执行命令：

show global status like 'Key\_reads';

show global status like 'Key\_read\_requests';

计算：

key\_cache\_miss\_rate = Key\_reads / Key\_read\_requests \* 100%(在0.1%与0.01%之间)

## 临时表

在数据库连接工具或登录数据库服务器后台执行命令：

show global status like 'Created\_tmp\_disk\_tables';

show global status like 'Created\_tmp\_tables';

计算：

Created\_tmp\_disk\_tables / Created\_tmp\_tables \* 100% <= 25%

## 表打开数

在数据库连接工具或登录数据库服务器后台执行命令：

show global status like 'Open\_tables';

show global status like 'Opened\_tables';

计算：

Open\_tables / Opened\_tables \* 100% >= 85%

## 进程使用

show global status like ‘Thread%’;

Threads\_created表示创建过的线程数，如果发现Threads\_created值过大的话，表明MySQL服务器一直在创建线程，这也是比较耗资源，可以适当增加配置文件中thread\_cache\_size值。

## 文件打开数

在数据库连接工具或登录数据库服务器后台执行命令:

show global status like 'Open\_files';

show VARIABLES like 'open\_files\_limit';

计算：

Open\_files / open\_files\_limit \* 100% <= 75%

# 附录(无)

# 附则(无)