**目录**

**第2** **章** **MySQL数据库服务日志2**

1.1 MySQL数据库服务常用日志文件知识

1.1.1 错误日志（ error log）介绍与调整

1.1.2 普通查询日志（ general query log）介绍与调整3

1.1.3 慢查询（ slow query log）介绍与调整

1.1.4 二进制日志（ binary log）介绍与调整

====================================================

1.1MySQL数据库服务常用日志文件知识

1.1.1 错误日志（error log）介绍与调整

①    错误日志（error log）介绍

MySQL的错误日志（error log）记录mysqld在启动/关闭/运行过程中遇到的错误信息；

②    错误日志（error log）实践

     方法一：在配置文件中调整方法，当然可以再启动时加入开机自启动参数

[mysqld\_safe]

log-error=/data/3306/mysql\_oldboy3306.err

方法二：启动mysql命令加入

mysql\_safe –defaults-file=/data/3306/my.cnf –log-eroor=/data/3306/mysql\_oldboy.err &

mysql> show variables like 'log\_error%';

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

| Variable\_name | Value                           |

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

| log\_error     | /data/3306/mysql\_oldboy3306.err |

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

1 row in set (0.00 sec)

学生遇到数据库起不来案例：（排查思路）

1. 把/data/3306/data 命令删除后重新建立data目录并加主属主
2. 把原来有的my.cnf mysql 这两个文件重新还原过
3. 重新用mysqld\_safe启动过
4. 改过my.cnf里面的server-id = 2
5. 配置文件里面的端口，路径

开始排查：

1. 先把日志文件备份并清空启动一下mysql服务然后查看日志文件报什么错
2. 然后mysql3306目录下面所有文件加上属主并是递归-R
3. 然后查看一下有没有这个ID号
4. 重启mysql服务
5. 查看服务进程是否启动

1.1.2 普通查询日志（general query log）介绍与调整

普通查询日志（general query log）介绍

普通查询日志（general query log）：记录客户端连接信息和执行的 SQL语句信息；

普通查询日志（general query log）调整

mysql> show variables like 'general\_log%';

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

| Variable\_name | Value |

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

| general\_log      | OFF   |

| general\_log-file|/data/3306/oldboy.log|

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

1 row in set (0.00 sec)

提示：企业一般是关闭的，因为性能问题

临时生效：

开启执行如下命令:

mysql> set global  general\_log = on;

Query OK, 0 rows affected (0.00 sec)

mysql> show variables like 'general\_log%';

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

| Variable\_name    | Value                    |

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

| general\_log      | ON                       |

| general\_log\_file | /data/3306/data/db02.log |

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

2 rows in set (0.00 sec)

永久生效

mysql> set global  general\_log\_file = "/data/3306/data/oldboy.log" ;

Query OK, 0 rows affected (0.00 sec)

mysql> show variables like 'general\_log%';

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

| Variable\_name    | Value                      |

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

| general\_log      | OFF                        |

| general\_log\_file | /data/3306/data/oldboy.log |

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

2 rows in set (0.00 sec)

或者

在配置文件中加入并重启

general\_log = on

general\_log\_file = /data/3306/data/oldboy.log

[mysqldump]

实际的普通查询日志示例：

[root@db02 data]# cat /data/3306/data/oldboy.log

/application/mysql-5.5.32/bin/mysqld, Version: 5.5.32-log (Source distribution). started with:

Tcp port: 3306  Unix socket: /data/3306/mysql.sock

Time                 Id Command    Argument

151221 16:09:25     3 Connect   rep@10.0.0.52 on

                    2 Connect Out       rep@10.0.0.52:3306

                    3 Query     SELECT UNIX\_TIMESTAMP()

                    3 Query     SHOW VARIABLES LIKE 'SERVER\_ID'

                    3 Quit

151221 16:09:26     4 Connect   root@localhost on

                    4 Query     show variables like 'general\_log%'

提示：高并发场景企业里普通查询日志一般是关闭（默认也是关闭的），因IO性能问题

1.1.3 慢查询（slow  query  log ）介绍与调整

慢查询日志（slow  query log）介绍

慢查询日志（ slow query log）：记录执行时间超出指定值（ long\_query\_time）的SQL 语句；

慢查询日志（slow query log）调整

long\_query\_time = 1

log-slow-queries = /data/3306/slow.log

log\_queries\_not\_using\_indexes

慢查询的设置，对数据库SQL的优化非常重要

[root@db02 ~]# egrep "quer"  /data/3306/my.cnf

long\_query\_time = 1

log\_queries\_not\_using\_indexes

log-slow-queries = /data/3306/slow.log

利用慢查询优化一套解决方案：

1.    开启慢查询

long\_query\_time = 1

log\_queries\_not\_using\_indexes

log-slow-queries = /data/3306/slow.log

2.    慢查询日志切割

[root@db02 3306]# mv slow.log slow.log.$(date +%F)

[root@db02 3306]# ls

data              mysql-bin.000006  mysql-bin.000014  mysqld.pid

my.cnf            mysql-bin.000007  mysql-bin.000015  mysql\_oldboy3306.err

mysql             mysql-bin.000008  mysql-bin.000016  mysql.sock

mysql-bin.000001  mysql-bin.000009  mysql-bin.000017  relay-bin.000016

mysql-bin.000002  mysql-bin.000010  mysql-bin.000018  relay-bin.index

mysql-bin.000003  mysql-bin.000011  mysql-bin.000019  relay-log.info

mysql-bin.000004  mysql-bin.000012  mysql-bin.000020 slow.log.2015-12-21

mysql-bin.000005  mysql-bin.000013  mysql-bin.index

[root@db02 3306]# mysqladmin -uroot -p123456 -S /data/3306/mysql.sock flush-log 刷新

[root@db02 3306]# ls

data              mysql-bin.000007  mysql-bin.000016      mysql.sock

my.cnf            mysql-bin.000008  mysql-bin.000017      relay-bin.000016

mysql             mysql-bin.000009  mysql-bin.000018      relay-bin.000017

mysql-bin.000001  mysql-bin.000010  mysql-bin.000019      relay-bin.index

mysql-bin.000002  mysql-bin.000011  mysql-bin.000020      relay-log.info

mysql-bin.000003  mysql-bin.000012  mysql-bin.000021      slow.log

mysql-bin.000004  mysql-bin.000013  mysql-bin.index      slow.log.2015-12-21

mysql-bin.000005  mysql-bin.000014  mysqld.pid

mysql-bin.000006  mysql-bin.000015  mysql\_oldboy3306.err

[root@db02 3306]#

编写脚本定时任务

[root@db02 scripts]# cat cut\_slow\_log.sh

cd /data/3306/ &&\

/bin/mv slow.log /backup/$(date +%F) &&\

mysqladmin -uroot -p123456 -S /data/3306/mysql.sock flush-log

mysqlsla /backup/$(date +%F)\_slow.log>new\_$(date +%F)\_slow.log

mail -s "$(date +%F)\_slow.log" 992747643@qq.com </backup/new\_$(date +%F)\_slow.log

[root@db02 scripts]# /bin/sh /server/scripts/cut\_slow\_log.sh  执行

加入定时任务

[root@db02 scripts]# tail -2 /var/spool/cron/root

#cut mysql slow log

00 00 \* \* \* /bin/sh /server/scripts/cut\_slow\_log.sh >/dev/null 2>&1

[root@db02 scripts]#

实用工具mysqlsla分析慢查询，定时发给运维人员信箱

1.1.4 二进制日志（binary log）介绍与调整

    二进制日志（binary log）介绍

二进制日志（binary log）：记录数据被修改的相关信息；

  二进制日志（binary log）调整

mysql> show variables like '%log\_bin%';

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

| Variable\_name                   | Value |

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

| log\_bin                         | ON    | 记录binlog

| log\_bin\_trust\_function\_creators      | OFF    |

| sql\_log\_bin                      | ON    |（临时不记录binlog增量恢复）

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

3 rows in set (0.00 sec)

[root@db02 3306]# grep log-bin /data/3306/my.cnf

log-bin = /data/3306/mysql-bin

log-bin作用

1. 记录更改SQL语句
2. 主从复制
3. 增量数据备份

临时不记录 binlog（增量恢复）

mysql> set session sql\_log\_bin = OFF;

Query OK, 0 rows affected (0.00 sec)

mysql> show variables like '%log\_bin%';

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

| Variable\_name                   | Value |

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

| log\_bin                         | ON    |

| log\_bin\_trust\_function\_creators | OFF   |

| sql\_log\_bin                     | OFF   |

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

3 rows in set (0.00 sec)

mysql> create database abc;

Query OK, 1 row affected (0.00 sec)

[root@db02 3306]# mysqlbinlog mysql-bin.000023|grep abc