

##### 学前须知 #####

优化:

数据表优化>sql优化>数据库参数配置>硬件资源优化

#####

## mysql物理文件与性能查询

### 1 数据库文件

MySQL数据库会在data目录下简历一个以数据库为名的文件夹,用来存储数据库中的表文件数据。不同的数据库引擎,每个表的扩展名也不一样,例如: MyISAM用“.MYD”作为扩展名,Innodb用“.ibd”,Archive 用“.arc”,CSV 用“.csv

myisam:

#### 1.1“.FRM”文件

!!!在mysql8.0已经取消,mysql的系统空间做了一个优化,所有表结构的存储不会再.FRM文件,都放在系统表空间里了。

无论是那种存储引擎,创建表之后就一定会生成一个以表明命名的'.frm'文件。frm文件主要存放与表相关的数据信息,主要包括表结构的定义信息。当数据库崩溃时,用户可以通过frm文件来恢复数据表结构。不包含索引和真实数据

#### 1.2“.MYD”文件

“.MYD”文件是MyISAM存储引擎专用,存放MyISAM表的数据。每一个MyISAM表都会有一个“.MYD”文件与之对应,同样存放于所属数据库的文件夹下,和“.frm”文件在一起。

#### 1.3“.MYI”文件(myisam data)

“.MYI”文件也是专属于MyISAM存储引擎的,主要存放MyISAM表的索引相关信息。对于MyISAM存储来说,可以被cache 的内容主要就是来源于“.MYI”文件中。每一个MyISAM表对应一个“.MYI”文件,存放于位置和“.frm”以及“.MYD”一样。

innodb:

#### 1.4“.ibd”文件与“.ibdata”文件

这两种文件都是存放Innodb数据的文件,之所以有两种文件来存放Innodb的数据(包括索引),是因为Innodb的数据存储方式能够通过配置来决定是使用共享表空间存放存储数据,还是独享表空间存放存储数据。独享表空间存储方式使用“.ibd”文件来存放数据,且每个表一个“.ibd”文件,文件存放在和MyISAM数据相同的位置。如果选用共享存储表空间来存放数据,则会使用ibdata文件来存放,所有表共同使用一个(或者多个,可自行配置)ibdata文件。

共享表空间(ibdata) (innodb\_file\_per\_table=0): 某一个数据库的所有的表数据,索引文件全部放在一个文件中。不能进行空间回收,100空间,即使删除数据,表空间也不会发生变化。

独享表空间(.ibd) (innodb\_file\_per\_table=1): 每一个表都将会生成以独立的文件方式来进行存储,每一个表都有一个.frm表描述文件,还有一个.ibd文件。其中这个文件包括了 单独一个表的数据 内容以及索引内容。能进行空间搜索。

两者对比

##### (1.) 共享表空间:

优点: 可以放表空间分成多个文件存放到各个磁盘上。数据和文件放在一起方便管理

缺点: 所有的数据和索引存放在一个文件中,多个表及索引在表空间中混合存储,这样对于一个表做了大量删除操作后表空间中将会有大量的空隙,特别是对 于统计分析,日值系统这类应用最不适合用共享表空间。

##### (2.) 独立表空间:

优点:

1. 每个表都有自己独立的表空间。
2. 每个表的数据和索引都会存在自己的表空间中。
3. 可以实现单表在不同的数据库中移动。
4. 空间可以回收

a) Drop table操作自动回收表空间,如果对于统计分析或是日值表,删除大量数据后可以通过:alter table TableName engine=innodb;回 缩不用的空间。

b) 对于使用独立表空间的表,不管怎么删除,表空间的碎片不会太严重的影响性能,而且还有机会处理。

缺点: 单表增加过大,如超过100 个G。 相比较之下,使用独占表空间的效率以及性能会更高一点 共享表空间和独立表空间之间的转换

show variables like "innodb\_file\_per\_table";

ON代表独立表空间管理,OFF代表共享表空间管理;

修改数据库的表空间管理方式 修改innodb\_file\_per\_table的参数值即可,但是修改不能影响之前已经使用过的共享表空间和独立表空间;

innodb\_file\_per\_table=1 为使用独占表空间 innodb\_file\_per\_table=0 为使用共享表空间

set global innodb\_file\_per\_table=1;(临时修改),永久修改在配置文件中添加内容

### 2.日志文件 (mysql中共六种日志文件,常说的是四种)

慢查询日志 事务日志 二进制日志文件 错误日志 中继日志 查询日志(后两种用的比较少)



## 2.1 查询日志

默认关闭, 影响性能

```
show variables like "%general_log%";
```

Variable_name	Value
general_log	OFF
general_log_file	D:\phpstudy\phpstudy_pro\Extensions\MySQL8.0.12\data\8I2BWZ073WJUZSJ.log

这两个值都可以通过配置文件添加

## 2.2 慢查询日志

默认不开启

记录指定超出某个时间执行的sql (2s, 3s)

作用: 项目中比较慢的查询进行捕获, 针对sql进行优化, 建议在项目完成之后开启

```
show variables like "%slow_query_log%";
```

Variable_name	Value
slow_query_log	OFF
slow_query_log_file	D:\phpstudy\phpstudy_pro\Extensions\MySQL8.0.12\data\8I2BWZ073WJUZSJ-slow.log

可以通过配置文件添加

```
slow_query_log = 1
```

```
slow_query_log_file=D:\phpstudy\phpstudy_pro\Extensions\MySQL8.0.12\data\8I2BWZ073WJUZSJ-slow.log
```

```
long_query_time=3
```

重启, 进行sql查询测试 `select sleep (5);`

## 2.3 错误日志

在mysql数据库中, 错误日志功能是默认开启的。

默认情况下错误日志大概记录以下几个方面的信息:

1. 服务器启动和关闭过程中的信息 (未必是错误信息, 如mysql如何启动InnoDB的表空间文件的、如何初始化自己的存储引擎的等等)
2. 服务器运行过程中的错误信息、事件调度器运行一个事件时产生的信息、在从服务器上启动服务器进程时产生的-信息

```
show global variables like "log_error";
```

Variable_name	Value
log_error	D:\phpstudy\phpstudy_pro\Extensions\MySQL8.0.12\data\8I2BWZ073WJUZSJ.err

!!!!!!我们可以根据错误日志查找操作过程的异常情况

## 2.4 二进制日志

也就是我们常说的binlog。会记录执行的sql (update delete insert create等), !!!!!启用该选项数据库性能降低1%, 但保障数据库完整性, 对于重要数据库值得以性能换完整。

默认情况下二进制日志功能是没有开启的, 启动可以配置log-bin[=file\_name]开启,

```
show global variables like "%log_bin%";
```

Variable_name	Value
log_bin	ON
log_bin_basename	D:\phpstudy\phpstudy_pro\Extensions\MySQL8.0.12\data\binlog
log_bin_index	D:\phpstudy\phpstudy_pro\Extensions\MySQL8.0.12\data\binlog.index
log_bin_trust_function_creators	OFF
log_bin_use_v1_row_events	OFF

每次重启都会生成一个二进制文件

作用

1. 以二进制形式记录更改数据库的SQL语句 (insert, update, delete, create, drop, alter等)。
2. 用于MySQL主从复制。
3. 增量数据备份及恢复

添加配置在my.cnf或者my.ini  
因为bug必须添加server-id数值随意指定

```
server-id=1  
log_bin=mysql-bin
```

binlog过大时如何处理？

- 1) 定时把二进制文件备份到数据库备份中
- 2) 定时对于往期1年进行选择删除或者更改

常用二进制查询语句：

- 1) show binary logs; 展示所有的二进制文件
- 2) 查看当前的日志 show master status;
- 3) 查询第一个二进制文件 show binlog events
- 4) 清空当前使用的二进制文件 reset master
- 5) 关闭当前使用的binary log,然后打开一个新的binary log文件,文件的序号加1 flush logs;

利用二进制文件进行数据恢复(mysqlbinlog文件所在位置,centos7下配置的mysql8.0一般在 /usr/bin)

#####

- 1 /etc/my.cnf 这是mysql的主配置文件
- 2 /var/lib/mysql mysql数据库的数据库文件存放位置
- 3 /var/log mysql数据库的日志输出存放位置
- 4 /usr/bin 各种工具组件所在位置
- 5 mysqlbinlog binlog.000013 > binlog.000013.txt 转换成txt文件,进行查看

#####

1)根据节点恢复：

```
mysqlbinlog --start-position=4719 --stop-position=6440 D:\MySQL8.0.12\data\binlog.000207 |mysql -uroot -p
```

2)根据时间恢复

```
mysqlbinlog --start-datetime="2020-03-28 22:22:22" --stop-position="2020-03-29 22:22:22" D:\MySQL8.0
```

```
.12\data\binlog.000207 |mysql -uroot -p
```

3)也可以直接利用指定binlog日志进行恢复

```
mysqlbinlog D:\MySQL8.0.12\data\binlog.000207 |mysql -uroot -p
```

## 2.5事务日志

- 1.重做日志redo
- 2.回滚日志undo

## 2.6中继日志

用来恢复数据（用于主从结构）

## 3.存储引擎

myisam 与 innodb

见图：

@鑫成尹