私塾在线《高级软件架构师实战培训阶段二》

——跟着cc学架构系列精品教程

SQL优化概述

- 基本的SQL优化的思路

1：截取SQL语句
2：识别有问题的SQL语句
3：确认语句执行
4：语句分析
5：语句优化
6：优化验证

截取SQL语句-1

- 全面查询日志

1：启用方式

（1）在mysql的配置文件中，设置如下：

```
general_log=1
general_log_file=/path/logfile
log_output=FILE
```

（2）也可以通过SQL语句来动态启用和禁用，还可以指定输出到表或者文件中：

```
set global general_log=1;
set global log_output='TABLE';
select * from mysql.general_log;
```

2：注意

全面日志提供所有连续的sql查询语句顺序，但不提供执行时间，在开发环境开启这个功能，可以很好的来审查SQL语句。**但永远不要在生产环境开启这个功能。**

截取SQL语句-2

- 慢查询日志

1：启用方式

（1）在mysql的配置文件中，设置如下：

```
slow_query_log=1
slow_query_log_file=/usr/local/mysql/logs/slowlog
long_query_time=0.2
log_output=FILE
```

- 进程列表

```
show full processlist;
```

- 引擎状态

show engine innodb status ，可以提供SQL语句的详细信息，比如：导致外键验证失败或者造成死锁的SQL语句。

- 其它

比如：开源项目sqlstats、MySql proxy等

识别有问题的SQL语句

- 通常关注

1：运行最慢的sql语句

2：运行较快，但是执行频率非常高的sql语句

- 识别方法

1：MySql自带的mysqldumpslow，如：

```
mysqldumpslow /usr/local/mysql/logs/slowlog
```

2：开源的pt-query-digest

下载地址：https://www.percona.com/software/mysql-tools/percona-toolkit

这个工具的输出结果有3个不同部分

（1）第一个部分提供了总体时间、查询频率、硬件故障细节等信息

（2）第二部分提供最差的不同查询的详细列表，按照总执行时间排序

（3）第三部分是执行时间的分布，以及具体的SQL语句

确认SQL语句执行

- 通常关注

1：环境

应该尽可能跟出问题的环境一样或者相似，也就是要让问题得以重现。

这样也利于优化改进后进行验证，看看优化是否有效果。

2：时间统计

采用同一个监控执行时间的标准，以利于发现问题，以及对比优化前后的效果

SQL语句分析-1

- 常用的SQL语句分析命令

1：explain：分析语句执行基于开销的优化器，以及被优化器考虑的访问策略等，常见的有

（1）select_type：表示table列引用的使用方式的类型：

a：simple：不包含子查询和其它复杂语法的简单查询

b：primary：复杂查询中的最外层的表

c：derived：表不是物理表

d：dependent：子查询

（2）table：表名、表的别名或者一个为查询产生的临时表的标识符

（3）type：表使用的连接方式，常用的有：

a：const：这个表最多只有一行匹配

b：system：表只有一行

c：eq_ref：有一行是为了每个之前确定的表而读取的

d：ref：所有具有匹配值的行都被用到

e：range：所有符合给定范围值的索引行都被用到

f：all：全表扫描

SQL语句分析-2

（4）possible_keys：优化器为查询选定的索引

（5）key：优化器选择使用的索引

（6）key_len：用于Sql语句的连接条件的键的长度，对于确认索引的有效性，以及多列索引中用到的列的数据很重要，常见的有：

key_len：4，表示int not null

key_len：5，表示int null

key_len：30，表示char(30) not null

key_len：32，表示varchar(30) not null

key_len：92，表示varchar(30) null

（7）ref：用来进行索引比较的列或者常量

（8）rows：优化器估计的所有存在于累计结果集中的行数目

（9）extra：优化器使用的额外信息

SQL语句分析-3

2：show create table 表名

展示表中列和索引定义的细节信息

3：show indexes from 表名

察看表的索引信息，其中的Cardinality表示索引中每一列唯一值的数量的估计值

4：show table status like 表名

察看表的底层大小以及表结构。对于innodb引擎，都是估计值。

5：show status命令

察看服务器当前内部状态信息，比如：show global status like ‘handler_read%’，用来查看是否使用了索引，以及使用索引读取了多少值

6：show variables命令

查看MySQL系统变量的当前值，比如：show variables like ‘%slow%’，用来查跟慢日志相关的变量的值

SQL语句优化

- 常用的SQL语句优化手段

1：尽量去除表连接操作
2：尽量减少操作到的列的数目
3：精简数据类型和约束条件，以改进表结构
4：合理的构建索引
5：在SQL中有意、合理的利用索引
6：去除重复索引
7：删除不用的索引
8：尽量减少sql语句要扫描的语句数量
9：确保on或者using子句上的列上有索引
10：确保group by和order by中的表达式只涉及表中的一个列
11：尽量明确写出要查询的列，少用select *
12：尽量不要在where里面使用 不等于 符号，或者是进行null值判断，这会导致全表扫描
13：尽量不要在where里面对字段进行函数式操作
14：用exist代替in
15：…… 太多太多的细节，这里就不去罗列了

- SQL语句优化实际是个综合性的工作

硬件服务器、MySQL配置、表结构、索引、临时表、SQL语句等一起综合考虑