

Mysql下优化SQL的一般步骤

<上一节

下一节>

关注我们:

微信公众号

新浪微博

读者QQ交流群: 664104694

C语言中文网推出辅导班啦，包括「C语言辅导班、C++辅导班、算法/数据结构辅导班」，全部都是一对一教学：一对一辅导 + 一对一答疑 + 布置作业 + 项目实践 + 永久学习。QQ在线，随时响应！

MySQL的安装

Linux下安装MySQL

Windows下安装Mysql

Windows下配置Mysql

Mysql安装注意事项

MySQL基础

Mysql存储引擎

Mysql数据类型

Mysql字符集

索引和锁

Mysql索引

锁机制和事务控制

安全和SQL Mode

使用SQL的安全问题

Mysql SQL Mode详解

SQL语句和表的优化

优化SQL的一般步骤

Mysql索引优化

大量插入数据时SQL的优化

insert语句的优化

group by语句的优化

order by语句的优化

join语句的优化

insert,update,delete的顺序

数据表的优化

锁的优化

其他优化措施

mysql连接池

减少对Mysql的访问

MySQL均衡负载

Mysql日志管理

通过show status和应用特点了解各种SQL的执行频率

通过SHOW STATUS可以提供服务器状态信息，也可以使用mysqladmin extended-status命令获得。SHOW STATUS可以根据需要显示session级别的统计结果和global级别的统计结果。

以下几个参数对Myisam和Innodb存储引擎都计数：

1. Com_select 执行select操作的次数，一次查询只累加1；
2. Com_insert 执行insert操作的次数，对于批量插入的insert操作，只累加一次；
3. Com_update 执行update操作的次数；
4. Com_delete 执行delete操作的次数。

以下几个参数是针对Innodb存储引擎计数的，累加的算法也略有不同：

1. Innodb_rows_read select查询返回的行数；
2. Innodb_rows_inserted执行Insert操作插入的行数；
3. Innodb_rows_updated 执行update操作更新的行数；
4. Innodb_rows_deleted 执行delete操作删除的行数。

通过以上几个参数，可以很容易的了解当前数据库的应用是以插入更新为主还是以查询操作为主，以及各种类型的SQL大致的执行比例是多少。对于更新操作的计数，是对执行次数的计数，不论提交还是回滚都会累加。

对于事务型的应用，通过Com_commit和Com_rollback可以了解事务提交和回滚的情况，对于回滚操作非常频繁的数据库，可能意味着应用编写存在问题。

此外，以下几个参数便于我们了解数据库的基本情况：

- Connections 试图连接Mysql服务器的次数
- Uptime 服务器工作时间
- Slow_queries 慢查询的次数

定位执行效率较低的SQL语句

可以通过以下两种方式定位执行效率较低的SQL语句：

- 可以通过慢查询日志定位那些执行效率较低的sql语句，用--log-slow-queries[=file_name]选项启动时，mysqld写一个包含所有执行时间超过long_query_time秒的SQL语句的日志文件。可以链接到管理维护中的相关章节。
- 慢查询日志在查询结束以后才纪录，所以在应用反映执行效率出现问题的时候查询慢查询日志并不能定位问题，可以使用show processlist命令查看当前MySQL在进行的线程，包括线程的状态，是否锁表等等，可以实时的查看SQL执行情况，同时对一些锁表操作进行优化。

通过EXPLAIN分析低效SQL的执行计划

通过以上步骤查询到效率低的SQL后，我们可以通过explain或者desc 获取MySQL如何执行SELECT语句的信息，包括select语句执行过程表如何连接和连接的次序。

explain可以知道什么时候必须为表加入索引以得到一个使用索引来寻找记录的更快的SELECT。

```
mysql> explain select sum(moneys) from sales a,companys b where a.company_id = b.id and a.year = 2006;
+-----+-----+-----+-----+-----+-----+-----+-----+
| select_type | table | type | possible_keys | key | key_len | rows | Extra |
+-----+-----+-----+-----+-----+-----+-----+-----+
| SIMPLE | b | index | PRIMARY | PRIMARY | 4 | 1 | Using index |
| SIMPLE | a | ALL | NULL | NULL | NULL | 12 | Using where |
+-----+-----+-----+-----+-----+-----+-----+-----+
2 rows in set (0.02 sec)
```

说明:

- select_type: select 类型
- table: 输出结果集的表
- type: 表示表的连接类型
 - ①当表中仅有一行是type的值为system是最佳的连接类型;
 - ②当select操作中使用索引进行表连接时type的值为ref;
 - ③当select的表连接没有使用索引时, 经常会看到type的值为ALL, 表示对该表进行了全表扫描, 这时需要考虑通过创建索引来提高表连接的效率。
- possible_keys: 表示查询时,可以使用的索引列.
- key: 表示使用的索引
- key_len: 索引长度
- rows: 扫描范围
- Extra: 执行情况的说明和描述

确定问题, 并采取相应的优化措施

经过以上步骤, 基本可以确认问题出现的原因, 可以根据情况采取相应的措施, 进行优化提高执行的效率。

例如上面的例子, 我们确认是对a表的全表扫描导致效率的不理想, 我们对a表的year字段创建了索引, 查询需要扫描的行数明显较少。

```
mysql> explain select sum(moneys) from sales a,companys b where a.company_id = b.id and a.year = 2006;
+-----+-----+-----+-----+-----+-----+-----+-----+
| select_type | table | type | possible_keys | key | key_len | rows | Extra |
+-----+-----+-----+-----+-----+-----+-----+-----+
| SIMPLE | b | index | PRIMARY | PRIMARY | 4 | 1 | Using index |
| SIMPLE | a | ref | year | year | 4 | 3 | Using where |
+-----+-----+-----+-----+-----+-----+-----+-----+
2 rows in set (0.02 sec)
```

C语言中文网推出辅导班啦, 包括「C语言辅导班、C++辅导班、算法/数据结构辅导班」, 全部都是**一对一教学**:
一对一辅导 + 一对一答疑 + 布置作业 + 项目实践 + 永久学习。QQ在线, 随时响应!

编程帮, 一个分享编程知识的公众号。跟着**站长**一起学习, 每天都有进步。
通俗易懂, 深入浅出, 一篇文章只讲一个知识点。
文章不深奥, 不需要钻研, 在公交、在地铁、在厕所都可以阅读, 随时随地涨姿势。
文章不涉及代码, 不烧脑细胞, 人人都可以学习。
当你决定关注「编程帮」, 你已然超越了90%的程序员!



微信扫描二维码关注

< 上一节

下一节 >

关注我们:

微信公众号

新浪微博

读者QQ交流群: 664104694

[关于C语言中文网](#) | [关于站长](#) | [如何才能完成一部教程](#) | [联系我们](#) | [网站地图](#) | [手机版网站](#)

精美而实用的网站，关注编程技术，追求极致，让您轻松愉快的学习。
Copyright ©2011-2015 biancheng.net, All Rights Reserved, 陕ICP备15000209号

