

# 干货 | MySQL 大表优化方案

小黑格子屋 昨天



当MySQL单表记录数过大时，增删改查性能都会急剧下降，可以参考以下步骤来优化：

## 单表优化

除非单表数据未来会一直不断上涨，否则不要一开始就考虑拆分，拆分会带来逻辑、部署、运维的各种复杂度，一般以整型值为主的表在千万级以下，字符串为主的表在五百万以下是没有太大问题的。而事实上很多时候MySQL单表的性能依然有不少优化空间，甚至能正常支撑千万级以上的数据量：

## 字段

- 1、尽量使用TINYINT、SMALLINT、MEDIUM\_INT作为整数类型而非INT，如果非负则加上UNSIGNED；
- 2、VARCHAR的长度只分配真正需要的空间；

- 3、使用枚举或整数代替字符串类型；
- 4、尽量使用TIMESTAMP而非DATETIME；
- 5、单表不要有太多字段，建议在20以内；
- 6、避免使用NULL字段，很难查询优化且占用额外索引空间；
- 7、用整型来存IP。

## 索引

- 1、索引并不是越多越好，要根据查询有针对性的创建，考虑在WHERE和ORDER BY命令上涉及的列建立索引，可根据EXPLAIN来查看是否用了索引还是全表扫描；
- 2、应尽量避免在WHERE子句中对字段进行NULL值判断，否则将导致引擎放弃使用索引而进行全表扫描；
- 3、值分布很稀少的字段不适合建索引，例如“性别”这种只有两三个值的字段；
- 4、字符字段只建前缀索引；
- 5、字符字段最好不要做主键；
- 6、不用外键，由程序保证约束；
- 7、尽量不用UNIQUE，由程序保证约束；
- 8、使用多列索引时主意顺序和查询条件保持一致，同时删除不必要的单列索引。

## 查询SQL

- 1、可通过开启慢查询日志来找出较慢的SQL；
- 2、不做列运算：SELECT id WHERE age + 1 = 10，任何对列的操作都将导致表扫描，它包括数据库教程函数、计算表达式等等，查询时要尽可能将操作移至等号右边；
- 3、sql语句尽可能简单：一条sql只能在一个cpu运算；大语句拆小语句，减少锁时间，一条大sql可以堵死整个库；
- 4、不用SELECT \*；
- 5、OR改写成IN：OR的效率是n级别，IN的效率是log(n)级别，in的个数建议控制在200以内；
- 6、不用函数和触发器，在应用程序实现；

- 7、避免%xxx式查询；
- 8、少用JOIN；
- 9、使用同类型进行比较，比如用'123'和'123'比，123和123比；
- 10、尽量避免在WHERE子句中使用 != 或 <> 操作符，否则将引擎放弃使用索引而进行全表扫描；
- 11、对于连续数值，使用BETWEEN不用IN：SELECT id FROM t WHERE num BETWEEN 1 AND 5；
- 12、列表数据不要拿全表，要使用LIMIT来分页，每页数量也不要太大。

## 引擎

目前广泛使用的是MyISAM和InnoDB两种引擎：

### MyISAM

MyISAM引擎是MySQL 5.1及之前版本的默认引擎，它的特点是：

- 1、不支持行锁，读取时对需要读到的所有表加锁，写入时则对表加排它锁；
- 2、不支持事务；
- 3、不支持外键；
- 4、不支持崩溃后的安全恢复；
- 5、在表有读取查询的同时，支持往表中插入新纪录；
- 6、支持BLOB和TEXT的前500个字符索引，支持全文索引；
- 7、支持延迟更新索引，极大提升写入性能；
- 8、对于不会进行修改的表，支持压缩表，极大减少磁盘空间占用。

### InnoDB

InnoDB在MySQL 5.5后成为默认索引，它的特点是：

- 1、支持行锁，采用MVCC来支持高并发；
- 2、支持事务；
- 3、支持外键；

- 4、支持崩溃后的安全恢复；
- 5、不支持全文索引。

总体来讲，MyISAM适合SELECT密集型的表，而InnoDB适合INSERT和UPDATE密集型的表。

### 系统调优参数

可以使用下面几个工具来做基准测试：

**sysbench**：一个模块化，跨平台以及多线程的性能测试工具。

**iibench-mysql**：基于 Java 的 MySQL、Percona、MariaDB 索引进行插入性能测试工具。

**tpcc-mysql**：Percona开发的TPC-C测试工具。

具体的调优参数内容较多，具体可参考官方文档，这里介绍一些比较重要的参数：

#### back\_log

backlog值指出在MySQL暂时停止回答新请求之前的短时间内多少个请求可以被存在堆栈中。也就是说，如果MySQL的连接数达到maxconnections时，新来的请求将会被存在堆栈中，以等待某一连接释放资源，该堆栈的数量即backlog，如果等待连接的数量超过backlog，将不被授予连接资源。可以从默认的50升至500。

#### wait\_timeout

数据库连接闲置时间，闲置连接会占用内存资源。可以从默认的8小时减到半小时。

#### maxuserconnection

最大连接数，默认为0无上限，最好设一个合理上限thread\_concurrency：并发线程数，设为CPU核数的两倍。

## **skipnameresolve**

禁止对外部连接进行DNS解析，消除DNS解析时间，但需要所有远程主机用IP访问。

## **keybuffersize**

索引块的缓存大小，增加会提升索引处理速度，对MyISAM表性能影响最大。对于内存4G左右，可设为256M或384M，通过查询 `show status like 'keyread%'`，保证 `keyreads / keyreadrequests`在0.1%以下最好。

## **innodbbufferpool\_size**

缓存数据块和索引块，对InnoDB表性能影响最大。通过查询 `show status like 'Innodbbufferpoolread%'`，保证  $(\text{Innodbbufferpoolreadrequests} - \text{Innodbbufferpoolreads}) / \text{Innodbbufferpoolreadrequests}$  越高越好。

## **innodbadditionalmempoolsize**

InnoDB存储引擎用来存放数据字典信息以及一些内部数据结构的内存空间大小，当数据库对象非常多的时候，适当调整该参数的大小以确保所有数据都能存放在内存中提高访问效率，当过小的时候，MySQL会记录Warning信息到数据库的错误日志中，这时就需要调整这个参数大小。

## **innodblogbuffer\_size**

InnoDB存储引擎的事务日志所使用的缓冲区，一般来说不建议超过32MB。

## **querycachesize**

缓存MySQL中的ResultSet，也就是一条SQL语句执行的结果集，所以仅仅只能针对select语句。当某个表的数据有任何任何变化，都会导致所有引用了该表的select语句在Query Cache中的缓存数据失效。所以，当我们的数据变化非常频繁的情况下，使用Query Cache可能会得不偿失。根据命中率  $(\text{Qcachehits} / (\text{Qcachehits} + \text{Qcache\_inserts}) * 100)$  进行调整，一般不建议太

大，256MB可能已经差不多了，大型的配置型静态数据可适当调大。

可以通过命令`show status like 'Qcache_ %'`查看目前系统Query cache使用大小。

### **readbuffersize**

MySql读入缓冲区大小。对表进行顺序扫描的请求将分配一个读入缓冲区，MySql会为它分配一段内存缓冲区。如果对表的顺序扫描请求非常频繁，可以通过增加该变量值以及内存缓冲区大小提高其性能。

### **sortbuffersize**

MySql执行排序使用的缓冲大小。如果想要增加ORDER BY的速度，首先看是否可以让MySQL使用索引而不是额外的排序阶段。如果不能，可以尝试增加sortbuffersize变量的大小。

### **readrndbuffer\_size**

MySql的随机读缓冲区大小。当按任意顺序读取行时(例如，按照排序顺序)，将分配一个随机读缓存区。进行排序查询时，MySql会首先扫描一遍该缓冲，以避免磁盘搜索，提高查询速度，如果需要排序大量数据，可适当调高该值。但MySql会为每个客户连接发放该缓冲空间，所以应尽量适当设置该值，以避免内存开销过大。

### **record\_buffer**

每个进行一个顺序扫描的线程为其扫描的每张表分配这个大小的一个缓冲区。如果你做很多顺序扫描，可能想要增加该值。

### **threadcachesize**

保存当前没有与连接关联但是准备为后面新的连接服务的线程，可以快速响应连接的线程请求而无需创建新的。

## table\_cache

类似于threadcachedsize，但用来缓存表文件，对InnoDB效果不大，主要用于MyISAM。

🔑 往期精选 🔑

[资源 | 探究 Java 线程状态及转化视频](#)

[十年前的你 VS 现在的你](#)

[干货 | 程序员面试 IT 公司的37个小贴士](#)

[又一互联网公司员工猝死，年仅25岁的应届毕业生...](#)

[传智招聘 | 千里之行，始于IT！](#)

来运：匠心零度

原文链接：

<https://segmentfault.com/a/1190000006158186>



一只黑

小黑是个好人，只推送有用的东西。



长按左侧二维码关注

微信ID: xhmsvip

小黑QQ: 3251175005

QQ群: 809453822

小黑微博: 小黑黑的格子屋

点赞的时候，请宠溺一点！

阅读 5063

36

精选留言

写留言



Chris丶

5

写的很好收藏了！



小刚

写的还可以吧，非专业人员相当用心了

1



动感光波  
学习学者



π 小欣

很详细，有没有Oracle的呀？



Galatea  
干货



孫先生~

今天面试刚问到

---