Java MySQL系列课程07

------ MySQL优化之MySQL Server优化

本文档包含以下内容：

1. MySQL查询缓存
2. 索引和数据缓存
3. MySQL线程缓存
4. 连接数量和超时时间

#### 前言：

对于MySQL Server端的优化，主要指的就是MySQL Server启动时加载的配置文件的配置项内容的优化（就是那个my.ini或者my.cnf），下面我们看看它的配置文件中有哪些是我们需要重点关注的优化参数。

#### 一 MySQL查询缓存：

**MySQL的查询缓存是把select查询语句上一次的查询结果记录下来放在缓存当中，下一次再查询相同内容的时候，直接从缓存中取出来就可以了，不用再进行一遍真正的SQL查询**。但是当两个select查询中间出现insert，update，delete语句的时候，查询缓存就会被清空。

查询缓存适用更新不频繁的表，因为当表更新频繁的话，查询缓存也总是被清空，过多的查询缓存的数据添加和删除，就会影响MySQL的执行效率，还不如每次都从磁盘上查来得快（缓存指的就是一块内存，内存I/O比磁盘I/O快很多）。

可以在MySQL上通过以下命令，来查看查询缓存的设置：

mysql> **show variables like '%query\_cache%';**

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

| Variable\_name | Value |

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

**| have\_query\_cache | YES | 表示MySQL Server是否支持查询缓存，此处YES为支持！**

| query\_cache\_limit | 1048576 |

| query\_cache\_min\_res\_unit | 4096 |

**| query\_cache\_size | 0 |** **表示查询缓存的大小，单位为M**

**| query\_cache\_type | OFF |**  **表示目前查询缓存目前未开启**

| query\_cache\_wlock\_invalidate | OFF |

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

6 rows in set, 1 warning (0.01 sec)

通过show status命令，可以查看MySQL查询缓存的使用状况，如下：

mysql> **show status like 'Qcache%';**

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

| Variable\_name | Value |

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

**| Qcache\_free\_blocks | 0 |**  **查询缓存中的空闲内存块的数目**

**| Qcache\_free\_memory | 0 |** **查询缓存的空闲内存总数**

| Qcache\_hits | 0 |

**| Qcache\_inserts | 0 |**  **被加入到缓存中的查询数目**

**| Qcache\_lowmem\_prunes | 0 |** **因为缺少内存而被从缓存中删除的查询数目**

| Qcache\_not\_cached | 0 |

**| Qcache\_queries\_in\_cache | 0 |** **在缓存中已注册的查询数目**

**| Qcache\_total\_blocks | 0 |**   **查询缓存中的块的总数目**

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

8 rows in set (0.01 sec)

**那查询缓存怎么开启呢**，很简单，可以通过set命令设置上面的缓存参数开启MySQL查询缓存功能，也可以找到MySQL的配置文件（windows是my.ini，linux是my.cnf），修改query\_cache\_type参数为1就可以了，然后重启MySQL Server就可以使用了！，如下：



再配置查询缓存的大小，单位为M，如下：



修改完成，重启MySQL Server，查询缓存生效。

#### 二 索引和数据缓存：

主要指的就是innodb\_buffer\_pool\_size配置项，从名字上就能看到，该配置项是针对InnoDB存储引擎起作用的，

这个参数**定义了InnoDB 存储引擎的表数据和索引数据的最大内存缓冲区大小**。innodb\_buffer\_pool\_size是**同时为数据块和索引块做缓存**，**这个值设得越高，访问表中数据需要的磁盘 I/O 就越少**。

 这个是配置文件上的配置项

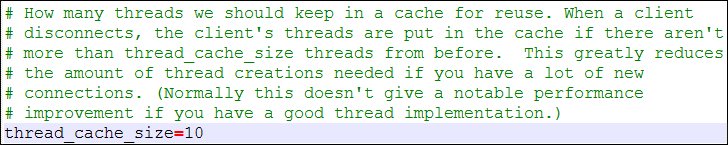
这个key\_buffer\_size只对MyISAM存储引擎起作用，还记得吗？MyISAM存储引擎采用非聚集索引，**所以它只缓存索引块**！

#### 三 **MySQL线程缓存**：

主要指配置文件中thread\_cache\_size配置项，第一节课的时候，给大家讲过MySQL Server网络模块采用经典的I/O复用（Java selector）+线程池模型（**注意：MySQL的源码是C语言实现的，这里的网络模型就相当于是Java的NIO Selector + 线程池模型**），之所以引入线程池，主要就是为了在业务执行的过程中，不会因为临时创建和销毁线程，造成系统性能降低，因为线程的创建和销毁是很耗费性能的。

所以，线程池就是在业务使用之前，先创建一组固定数量的线程，等待事件发生，当有SQL请求到达MySQL Server的时候，在线程池中取一个线程来执行该SQL请求就可以了，执行完成后，不销毁线程，而是把线程再归还到线程池中，等待下一次任务的处理（当然MySQL会根据用户量，自动加大线程池的数量的）。

如下配置：



配置完thread\_cache\_size，**重启MySQL Server服务生效**。

#### 四 连接数量和超时时间：

MySQL Server作为一个服务器，可以设置客户端的最大连接量和连接超时时间，如果数据库连接统计数量比较大，这两个参数的值需要设置大一些。

mysql> show variables like '%connect%';

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

| Variable\_name | Value

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

| character\_set\_connection | utf8

| collation\_connection | utf8\_general\_c

| connect\_timeout | 10

| disconnect\_on\_expired\_password | ON

| init\_connect |

| max\_connect\_errors | 100

| max\_connections | 151 MySQL Server支持的最大连接数量

| max\_user\_connections | 0

| performance\_schema\_session\_connect\_attrs\_size | 512

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

在配置文件（my.cnf或my.ini）最下面，添加配置：

max\_connections=2000，然后重启MySQL Server，设置生效。

mysql> show global variables like '%timeout%';

| wait\_timeout | 28800 |

MySQL Server对于超时未通信的连接，进行主动关闭操作

设置超时时间，超过设置时间没有请求就主动断开，单位是秒，在配置文件中添加配置：wait\_timeout = 600。