Atitit 数据库查询缓存mysql query cache

目录

[1. 性能提升约20%-30% 1](#_Toc12750)

[1.1. 开启必须要修改配置文件 1](#_Toc15683)

[1.2. 显示缓存状态 1](#_Toc22616)

[1.3. 调整参数化， 1](#_Toc5225)

[1.4. 评估缓存命中率与缓存状态 2](#_Toc10041)

# 性能提升约20%-30%

短期大批量查询还是很有效的，。短期内更新类操作会比较少

## 开启必须要修改配置文件

query\_cache\_type=1

## 显示缓存状态

show global variables like '%query\_cache%';

show global status like '%qcache%'

## 调整参数化，

可以根据缓存状态来调整参数

-- cache

set global query\_cache\_type=1;

set global query\_cache\_limit=10234; -- 10M per rsult size limit

set global query\_cache\_size=800123456;

## 评估缓存命中率与缓存状态

在开奖测试中，缓存命中率约50%

Qcache\_hits 缓存命中次数

Qcache\_inserts 3479418

show global status like '%qcache%'

Qcache\_free\_memory 291983080

Qcache\_hits 17403131

Qcache\_inserts 3479418

Qcache\_lowmem\_prunes 37775

Qcache\_not\_cached 8465586

Qcache\_queries\_in\_cache 158587

Qcache\_total\_blocks 318561

# ****3.失效机制：****

    当后端任何一个表的一条数据，索引，结构发生变化时，就会将与此表关联的query chache失效，并且释放内存。所以对于数据变化频繁的sql就不要cache了。那样不但不会提高性能还能得到相反的结果，因为每次多了查询缓存的操作。

    这里要指出的是，这种失效机制并不科学，因为有些表的改动并不会导致结果集的改变。但是这种方法简单，开销也比较小。

query\_cache\_limit：允许 Cache 的单条 Query 结果集的最大容量，默认是1MB，超过此参数设置的 Query 结果集将不会被 Cache

1. ****Query Cache 处理子查询：****  
      Query Cache 是以客户端请求提交的Query 为对象来处理的，只要客户端请求的是一个Query，无论这个 Query 是一个简单的单表查询还是多表 Join，亦或者是带有子查询的复杂 SQL，都被当作成一个Query，不会被分拆成多个Query 来进行Cache。所以，存在子查询的复杂Query 也只会产生一个Cache对象，子查询不会产生单独的Cache内容。UNION[ALL] 类型的语句也同样如此。

# 优化查询缓存

分拆多个子查询 让其可以进入cache

变化频繁的表使用内存表，雾化试图来cache

改写sql ，替换orm生产的sql为手工sql，方便分析和提前填充cache