

MySQL 查询缓存

提高单台节点的性能无外乎就那么几种方式，缓存是首当其冲的，因为内存的性能比磁盘高的太多。缓存也是一种典型的空间换时间的策略

缓存的实现也有太多的方式，从静态页面缓存到服务端动态缓存，再到数据库级别缓存等等。随着大数据的到来也产生了太多太多的内存数据库。

缓存的工具也多种多样，随口一说就能说出很多，像 Redis, Memcached, Hbase 等，就连 MySQL 也有一个 memory 引擎数据就是直接存储到内存中，所以缓存无处不在。

缓存的形式也多种多样，上面说的算是比较传统的缓存，其实各种的消息队列何尝又不是一种缓存机制。用到缓存的还有一个对于我们开发者都比较熟悉的，就是谷歌的低成本之道，据说谷歌最初的时候为了降低成本，提高性能，所有的机器中都没有装有硬盘，而是直接使用内存，数据靠多个节点的冗余来避免数据丢失。

缓存也是有缺陷，既然是缓存，那么针对的对象都是一些不变或者变化不大的对象，如果数据是经常变化的那么对其缓存反而适得其反。

说了那么多了，下面就说说 MySQL 的查询缓存吧，MySQL 的查询缓存是 MySQL 内置的一种缓存机制，可以针对 sql 进行缓存。比如我们发送 `select * from mysql.user` 这么一个查询，MySQL 首先检索内存中是否有数据并且数据是否过期，如果没有数据或者数据已经过期就去数据库中查找，如果有数据并且没有过期就直接返回数据。对于 sql 的匹配规则非常简单，就是字符串的比较，只要字符串相同，那么就认为是同一个查询。这里的字符串相同并不是表示 sql 语义相同，而是查询的 sql 字符串相同，空格也不行。MySQL 的缓存是对全部的 sql 有效的，也就是说一旦开启了查询缓存，那么对所有的 sql 查询默认都是开启的。我们会有很多变化的数据，其实是不希望开启查询缓存的，这个 MySQL 也给我们想到了，就是在 sql 中加入 `sql_no_cache` 比如，`select sql_no_cache * from mysql.user` 这样，MySQL 就会绕过缓存直接从库里查找数据。另外还有一个需要注意的，比如我们在 sql 中使用了 `now()` 这样的函数，MySQL 是不会给我们缓存的。

先来看一下有没有开启查询缓存

```
mysql> show variables like '%query%';
```

Variable_name	Value
binlog_rows_query_log_events	OFF
ft_query_expansion_limit	20
have_query_cache	YES
long_query_time	10.000000
query_alloc_block_size	8192
query_cache_limit	1048576
query_cache_min_res_unit	4096
query_cache_size	0
query_cache_type	OFF
query_cache_wlock_invalidate	OFF
query_prealloc_size	8192
slow_query_log	ON
slow_query_log_file	DESKTOP-80B3T3T-slow.log

```
13 rows in set, 1 warning (0.01 sec)
```

开启查询缓存的方式也非常简单，在 my.cnf 配置文件中设置上面的关于查询缓存的变量就可以了，下面具体说说每个变量的含义。

query_cache_type 是否开启查询缓存，0 表示不开启查询缓存，1 表示始终开启查询缓存（不要缓存使用 sql_no_cache），2 表示按需开启查询缓存（需要缓存使用 sql_cache）。

query_cache_size 给缓存分配的最大内存空间

对于查询缓存的一些操作。

- 1、FLUSH QUERY CACHE; // 清理查询缓存内存碎片。
- 2、RESET QUERY CACHE; // 从查询缓存中移出所有查询。
- 3、FLUSH TABLES; // 关闭所有打开的表，同时该操作将会清空查询缓存中的内容。

缓存虽然能够大幅度提高性能，但用起来也要慎重，一旦用不好，反而会适得其反。