

[MySQL FAQ]系列 — 线上环境到底要不要开启query cache

适用场景：

Query Cache（查询缓存，以下简称QC）存储SELECT语句及其产生的数据结果，特别适用于：频繁提交同一个语句，并且该表数据变化不是很频繁的场景，例如一些静态页面，或者页面中的某块不经常发生变化的信息。QC有可能会从InnoDB Buffer Pool或者MyISAM key buffer里读取结果。

InnoDB缓存池

由于QC需要缓存最新数据结果，因此表数据发生任何变化（INSERT、UPDATE、DELETE或其他可能产生数据变化的操作），都会导致QC被刷新。

根据MySQL官方的测试，QC的优劣分别是：

- 1、如果对一个表执行简单的查询，但每次查询都不一样的话，打开QC后，性能反而下降了13%左右。但通常实际业务中，通常不会只有这种请求，因此实际影响应该比这
- 2、如果对一个只有一行数据的表进行查询，则可以提升238%，这个效果还是非常不错的。

读多写少

因此，如果是在一个更新频率非常低而只读查询频率非常高的场景下，打开QC还是比较有优势的，其他场景下，则不建议使用。而且，QC一般也维持在100MB以内就够了，没必要设置超过数百MB。

QC严格要求2次SQL请求要完全一样，包括SQL语句、连接的数据库、协议版本、字符集等因素都会影响，下面几个例子中的SQL会被认为是完全不一样而不会使用同一个QC内存块：

```
mysql> set names latin1; SELECT * FROM table_name;
mysql> set names latin1; select * from table_name;
mysql> set names utf8; select * from table_name;
```

SQL 语句关键字大小写

编码不同

此外，QC也不适用于下面几个场景：

- 1、子查询或者外层查询；
- 2、存储过程、存储函数、触发器、event中调用的SQL，或者引用到这些结果的；
- 3、包含一些特殊函数时，例如：BENCHMARK()、CURDATE()、CURRENT_TIMESTAMP()、NOW()、RAND()、UUID()等等；
- 4、读取mysql、INFORMATION_SCHEMA、performance_schema 库数据的；
- 5、类似SELECT...LOCK IN SHARE MODE、SELECT...FOR UPDATE、SELECT...INTO OUTFILE/DUMPFIL、SELECT...WHERE IS NULL等语句；
- 6、SELECT执行计划用到临时表（TEMPORARY TABLE）；
- 7、未引用任何表的查询，例如 SELECT 1+1 这种；
- 8、产生了 warnings 的查询；
- 9、SELECT语句里加了 SQL_NO_CACHE 关键字；

更加奇葩的是，MySQL在从QC中取回结果前，会先判断执行SQL的用户是否有全部库、表的SELECT权限，如果没有，则也不会使用QC。

相比下面这个，其实上面所说的都不重要。

更新频繁时，性能肯定不会好

最为重要的是，在MySQL里QC是由一个全局锁在控制，每次更新QC的内存块都需要进行锁定。

例如，一次查询结果是20KB，当前 `query_cache_min_res_unit` 值设置为4KB（默认值就是4KB，可调整），那么本次查询结果共需要分为5次写入QC，每次都要锁定，可见其成本有多高。

我们可以通过 **PROFILING** 功能来查看 QC 相关的一些锁竞争，例如像下面这样的：

- Waiting for query cache lock
- Waiting on query cache mutex

或者，也可以通过执行 **SHOW PROCESSLIST** 来看线程的状态，例如：

- checking privileges on cached query

检查用户是否有权限读取QC中的结果集

- checking query cache for query

检查本次查询结果是否已经存储在QC中

- invalidating query cache entries

由于相关表数据已经修改了，因此将QC中的内存记录被标记为失效

- sending cached result to client

从QC中，将缓存后的结果返回给客户程序

- storing result in query cache

将查询结果缓存到QC中

如果可以频繁看到上述几种状态，那么说明当前QC基本存在比较重的竞争。

说了这么多废话，其实核心要点就一个：

如果线上环境中99%以上都是只读，很少有更新，再考虑开启QC吧，否则，就别开了。

关闭方法很简单，有两种：

query_cache_type = 0 和 query_cache_size = 0

- 1、同时设置选项 `query_cache_type = 0` 和 `query_cache_size = 0`；
- 2、如果用源码编译MySQL的话，编译时增加参数 `--without-query-cache` 即可；

延伸阅读：

<http://www.dbasquare.com/kb/how-query-cache-can-cause-performance-problems/>

<http://www.percona.com/blog/2012/09/05/write-contentions-on-the-query-cache/>

<http://dev.mysql.com/doc/refman/5.6/en/query-cache.html>