

Mysql内存中处处都存在缓存，这里介绍一下几种显示的缓存优化。

1. 查询缓存优化：

查询缓存不仅把查询语句结构缓存起来，还将查询结果缓存起来。一段时间内，如果是同样的sql，则直接从缓存中读取数据，提高查找数据的效率。但当缓存中的数据和硬盘中的数据不一致时，缓存就会失效。

```
mysql> show variables like '%query_cache%'
-> ;
```

Variable_name	Value
have_query_cache	YES
query_cache_limit	1048576
query_cache_min_res_unit	4096
query_cache_size	16777216
query_cache_type	OFF
query_cache_wlock_invalidate	OFF

```
6 rows in set (0.00 sec)
```

have_query_cache：是否支持缓存。query_cache_limit：如果某条select语句超过了值，则不会被添加到查询缓存中。

query_cache_min_res_unit：每次申请空间的大小。query_cache_type：OFF表示查询缓存是关闭的。ON表示查询总是先到查询缓存中去找，除非在select语句中包含sql_no_cache选项。

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

Variable_name	Value
Qcache_free_blocks	1
Qcache_free_memory	16760152
Qcache_hits	0
Qcache_inserts	0
Qcache_lowmem_prunes	0
Qcache_not_cached	15
Qcache_queries_in_cache	0
Qcache_total_blocks	1

```
8 rows in set (0.00 sec)
```

Qcache_free_blocks:表示查询缓存中处以重现状态的内存块数，如果Qcache_free_blocks的值较大，则意味着查询缓存中碎片较多。

Qcache_free_memeory:表示当前mysql服务实例的查询缓存还有多少可用内存。

Qcache_hits:表示使用查询缓存的次数，改值会依次增加。如果Qcache_hits比较大，则说明查询使用的非常频繁，此时需要增加缓存。。。。

2.结果集缓存

结果集缓存是会话缓存，mysql客户机成功连接服务器之后。mysql服务器会为每个mysql客户机保留结果集缓存。缓存mysql客户机连接线程的连接信息以及缓存返回mysql客户机的结果集信息，当mysql客户机向服务器发送select语句时，mysql将select语句的执

行结果暂存到结果集缓存中。结果集的缓存大小由net_buffer_length数值定义。

```
mysql> show variables like 'net_buffer_length';
+-----+-----+
| Variable_name | Value |
+-----+-----+
| net_buffer_length | 16384 |
+-----+-----+
1 row in set (0.00 sec)

mysql> show variables like 'max_allowed_packet';
+-----+-----+
| Variable_name | Value |
+-----+-----+
| max_allowed_packet | 16777216 |
+-----+-----+
1 row in set (0.01 sec)
```

3. 排序缓存

1. 普通排序缓存，如果客户机向服务端发送的sql语句中含有设计排序的order by或者group by子句。mysql就会选择相应的排序算法。在普通排序索引上进行排序，提升排序速度。普通排序索引的大小由sort_buffer_size参数定义，如果要提升排序速度，首先应该添加合适的索引，则后应该增大排序索引的缓存sort_buffer_size

```
mysql> select @@global.sort_buffer_size / 1024
-> ;
+-----+
| @@global.sort_buffer_size / 1024 |
+-----+
| 1024.0000 |
+-----+
1 row in set (0.00 sec)
```

2. MyISAM排序索引

为了加快重建索引的效率，MyISAM提供了排序索引缓存用于实现索引的排序工作，这些方法都是尽量使排序的工作在内存中完成。MyISAM排序缓存的大小由myisam_sort_buffer_size定义。索引重建后，该缓存立马释放。

当排序的缓存超过myisam_sort_buffer_size的阈值时，此时就需要在临时文件中完成索引字段的排序工作，外存临时文件的大小由MyISAM_max_sort_file_size参数设定。索引重建后，临时文件立即删除。

```
mysql> select @@global.myisam_sort_buffer_size / 1024
-> ;
+-----+
| @@global.myisam_sort_buffer_size / 1024 |
+-----+
| 8192.0000 |
+-----+
1 row in set (0.00 sec)

mysql> select @@global.myisam_max_sort_file_size / 1024;
+-----+
| @@global.myisam_max_sort_file_size / 1024 |
+-----+
| 9007199254739967.7734 |
+-----+
1 row in set (0.00 sec)
```

3、InnoDB 排序缓存

和MyISAM引擎类似，当执行alter table、create index 创建索引是，InnoDB提供了3个InnoDB排序缓存用于实现索引的排序，每个缓存的大小由innodb_sort_buffer_size定义。

```
mysql> select @@global.innodb_sort_buffer_size/1024;
+-----+
| @@global.innodb_sort_buffer_size/1024 |
+-----+
| 1024.0000 |
+-----+
```

<https://www.ib51.net/article/108177.htm> 还有部分内容