Atitit mysql 查询性能 提升

索引

**query\_cache\_size：缓存MySQL中的ResultSet，也就是一条SQL语句执行的结果集，所以仅仅只能针对select语句。当某个表的数据有任何任何变化，都会导致所有引用了该表的select语句在Query Cache中的缓存数据失效。所以，当我们的数据变化非常频繁的情况下，使用Query Cache可能会得不偿失。根据命中率(Qcache\_hits/(Qcache\_hits+Qcache\_inserts)\*100))进行调整，一般不建议太大，256MB可能已经差不多了，大型的配置型静态数据可适当调大.  
可以通过命令show status like 'Qcache\_%'查看目前系统Query catch使用大小**

**read\_buffer\_size：MySql读入缓冲区大小。对表进行顺序扫描的请求将分配一个读入缓冲区，MySql会为它分配一段内存缓冲区。如果对表的顺序扫描请求非常频繁，可以通过增加该变量值以及内存缓冲区大小提高其性能**

**sort\_buffer\_size：MySql执行排序使用的缓冲大小。如果想要增加ORDER BY的速度，首先看是否可以让MySQL使用索引而不是额外的排序阶段。如果不能，可以尝试增加sort\_buffer\_size变量的大小**

**read\_rnd\_buffer\_size：MySql的随机读缓冲区大小。当按任意顺序读取行时(例如，按照排序顺序)，将分配一个随机读缓存区。进行排序查询时，MySql会首先扫描一遍该缓冲，以避免磁盘搜索，提高查询速度，如果需要排序大量数据，可适当调高该值。但MySql会为每个客户连接发放该缓冲空间，所以应尽量适当设置该值，以避免内存开销过大。**

**record\_buffer：每个进行一个顺序扫描的线程为其扫描的每张表分配这个大小的一个缓冲区。如果你做很多顺序扫描，可能想要增加该值**

**thread\_cache\_size：保存当前没有与连接关联但是准备为后面新的连接服务的线程，可以快速响应连接的线程请求而无需创建新的**

**table\_cache：类似于thread\_cache\_size，但用来缓存表文件，对InnoDB效果不大，主要用于MyISAM**