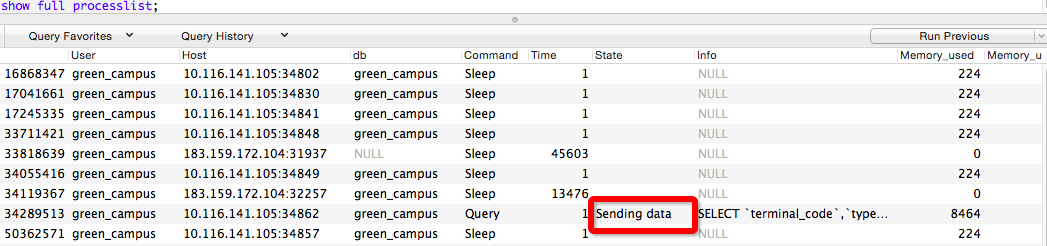
mysql的分析与优化

1. **分析线程**

执行sql语句show full processlist

* 1. State是Sending data



**原因：**

sending data的含义是“收集+发送数据”，如果通过索引查询出来的数据。如果有的数据不在索引列里面就需要重新到“数据行”中将需要返回的数据读取出来给客户端，这种一般是建了索引没用上，或者是存在大字段的数据引起

**解决：**

**重建索引，索引优化参考文档**

1. **优化my.cnf**

**#innodb引擎独立表空间**

innodb是使用表空间进行数据存储的，配置变量是innodb\_file\_per\_table。

set innodb\_file\_per\_table = on。

设置ON: 独立表空间，存储时在指定数据库目录会产生【表名】.ibd。

set innodb\_file\_per\_table = off。

设置OFF:系统表空间，存储时在全局数据库目录会产生ibdata1,ibdata2....等等。

在设置时建议采用 独立表空间，独立表空间的性能更优，系统表空间在数据变大时，会产生大量的磁盘碎片。

**#my.cnf的配置清单**

**[mysqld]**

port = 3306

socket = /tmp/mysql.sock

datadir = /usr/local/mysql/var #数据库存储目录

default-storage-engine=INNODB #数据库默认引擎

skip-external-locking #跳过外部锁定

skip-name-resolve #对外部连接进行DNS连接

max\_connections = 3000 #默认为100，对于数据库访问，操作频繁的系统建议加大值。但不能超过16384

key\_buffer\_size = 16M #这个值决定了索引的处理速度，并且只对myisam表起作用，

table\_open\_cache = 64 #Open\_tables/Opened\_tables >= 0.85；Open\_tables/table\_open\_cache <= 0.95

sort\_buffer\_size = 1M #增加sort\_buffer\_size值来加速ORDER BY或者GROUP BY的操作

net\_buffer\_length = 256K #

read\_buffer\_size = 1M

read\_rnd\_buffer\_size = 1M

myisam\_sort\_buffer\_size = 128M

log-bin=mysql-bin

binlog\_format=mixed

server-id=1

innodb\_buffer\_pool\_size = 4G

innodb\_additional\_mem\_pool\_size = 64M

innodb\_log\_file\_size = 256M

innodb\_log\_buffer\_size = 16M

innodb\_flush\_log\_at\_trx\_commit = 2 #这个值很重要，设置成为2表示事务不需要实时刷新到磁盘缓存日志

innodb\_lock\_wait\_timeout = 120

**[mysqldump]**

quick

max\_allowed\_packet = 32M

**[myisamchk]**

key\_buffer\_size = 1024M

sort\_buffer\_size = 1024M

read\_buffer = 16M

write\_buffer = 16M