线上有个MySQL实例，存在严重的复制延迟问题，原因出乎意料。

线上有个MySQL 5.7版本的实例，从服务器延迟了3万多秒，而且延迟看起来好像还在加剧。

MySQL版本

Server version:     5.7.18-log MySQL Community Server (GPL)

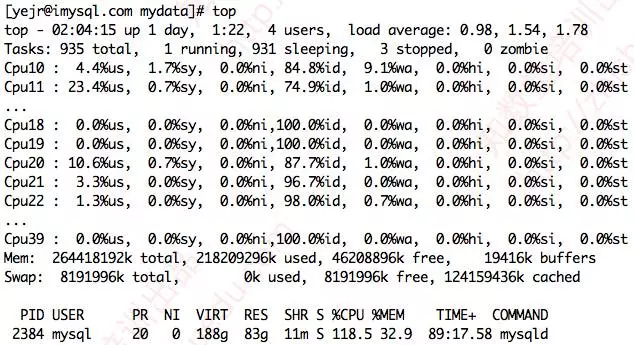
看下延迟状况

yejr@imysql.com:mysql3306.sock : (none) > show slave status\G              Master\_Log\_File: mysql-bin.013225          Read\_Master\_Log\_Pos: 1059111551        Relay\_Master\_Log\_File: mysql-bin.013161          Exec\_Master\_Log\_Pos: 773131396                  Master\_UUID: e7c35a95-ffb1-11e6-9620-90e2babb5b90

我们看到，**binlog文件落后了64个**，相当的夸张。

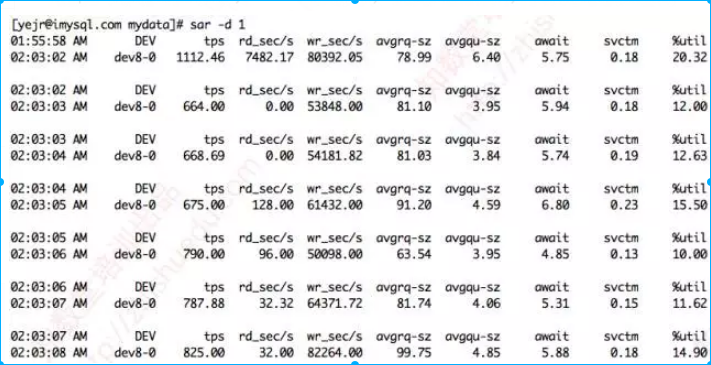
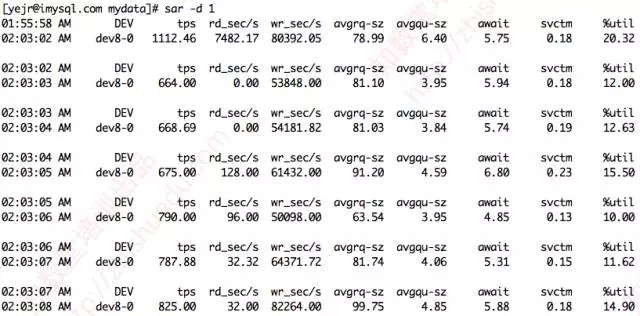
MySQL 5.7不是已经实现并行复制了吗，怎么还会延迟这么厉害？

先检查系统负载。



看到**mysqld进程其实负载还好，不算太高，也不存在严重的SWAP等问题**。

再看I/O子系统负载，没看到这方面存在瓶颈（**await\svctm\%util都不高**）。



再看mysqld进程的CPU消耗。

**虽然mysqld进程的CPU消耗总是超过100%，不过也不算太高。**

再检查MySQL复制现场，确认了**几个频繁更新的表都有主键，以及必要的索引**。相应的DML操作也几乎都是基于主键或唯一索引条件执行的，**排除无主键、无合理索引方面的因素**。

最后只能祭出**perf top神**器了。

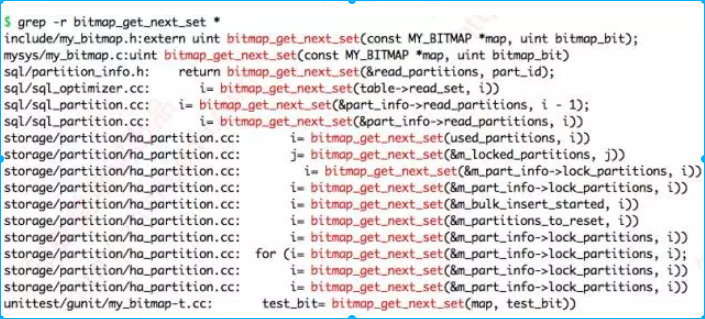
perf top -p `pidof mysqld`

看到perf top最后的报告是这样的

Samples: 107K of event 'cycles', Event count (approx.): 29813195000                                                                                                                               Overhead  Shared Object        Symbol                                                                                                                                                              56.19%  mysqld               [.] **bitmap\_get\_next\_set**                                                                                                                                            16.18%  mysqld               [.] **build\_template\_field**                                                                                                                                             4.61%  mysqld               [.] ha\_innopart::try\_semi\_consistent\_read                                                                                                                           4.44%  mysqld               [.] dict\_index\_copy\_types                                                                                                                                           4.16%  libc-2.12.so         [.] \_\_memset\_sse2                                                                                                                                                   2.92%  mysqld               [.] ha\_innobase::build\_template

我们看到， **bitmap\_get\_next\_set** 这个函数调用占到了 56.19%，非常高，其次是 **build\_template\_field** 函数，占了 16.18%。

经过检查MySQL源码并请教MySQL内核开发专家，最后确认这两个函数跟启用表分区有关系。



查询下当前实例有多少个表分区：

yejr@imysql.com:mysql3306.sock : (none) > select count(\*) from partitions where partition\_name is not null; +----------+ | count(\*) | +----------+ |    32128 | +----------+ 1 row in set (11.92 sec)

额滴神啊，竟然有3万多个表分区，难怪上面那两个函数调用那么高。

这个业务数据库几个大表采用每天一个分区方案，而且把直到当年年底所有分区也都给提前创建好了，所以才会有这么多。

不过，虽然有这么多表分区，在master服务器上却不存在这个瓶颈，看起来是在主从复制以及大量表分区的综合因素下才有这个瓶颈，最终导致主从复制延迟越来越严重。

知道问题所在，解决起来就简单了。**把到下个月底前用不到的表分区全部删除，之后约只剩下1.6万个分区。重启slave线程，问题解决，主从复制延迟很快就消失了**。