Mysql内核深度优化

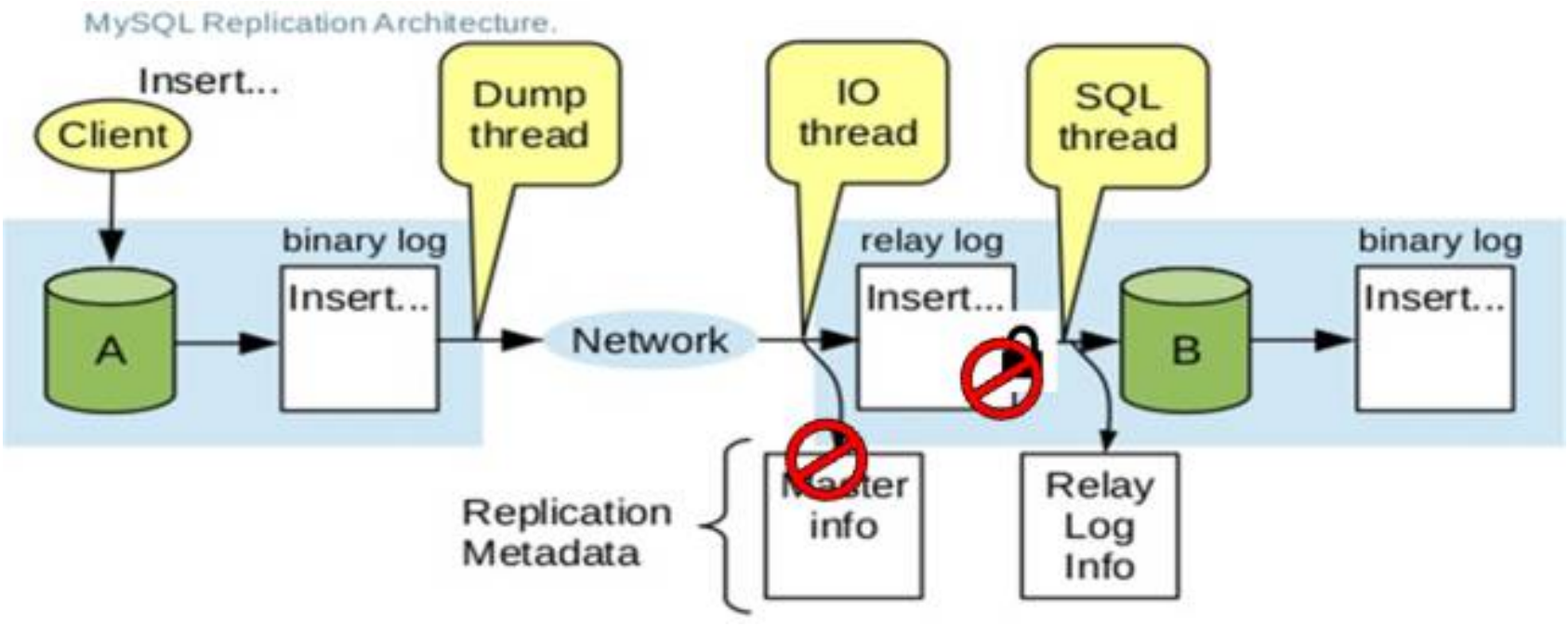
> 简怀兵，腾讯云数据库高级工程师，负责腾讯云CDB内核及基础设施建设；先后供职于Thomson Reuters和YY等公司，PTimeDB作者，曾获一项发明专利；从事mysql内核开发工作8年，具有丰富的优化经验；在分布式存储等领域有较丰富经验。

早期的CDB主要基于开源的Oracle MySQL分支，侧重于优化运维和运营的OSS系统。在腾讯云，因为用户数的不断增加，对CDB for MySQL提出越来越高的要求，腾讯云CDB团队针对用户的需求和业界发展的技术趋势，对CDB for MySQL分支进行深度的定制优化。优化重点围绕内核性能、内核功能和外围OSS系统三个维度展开，具体的做法如下：

一.内核性能的优化

由于腾讯云上的DB基本都需要跨园区灾备的特性，因此CDB for MySQL的优化主要针对主从DB部署在跨园区网络拓扑的前提下，重点去解决真实部署环境下的性能难题。经过分析和调研，我们将优化的思路归纳为：“消除冗余I/O、缩短I/O路径和避免大锁竞争”。以下是内核性能的部分案例：

1.主备DB间的复制优化



问题分析

如上图所示，在原生MySQL的复制架构中，Master侧通过Dump线程不断发

送Binlog事件给Slave的I/O线程，Slave的I/O线程在接受到Binlog事件后，有两个主要的动作：

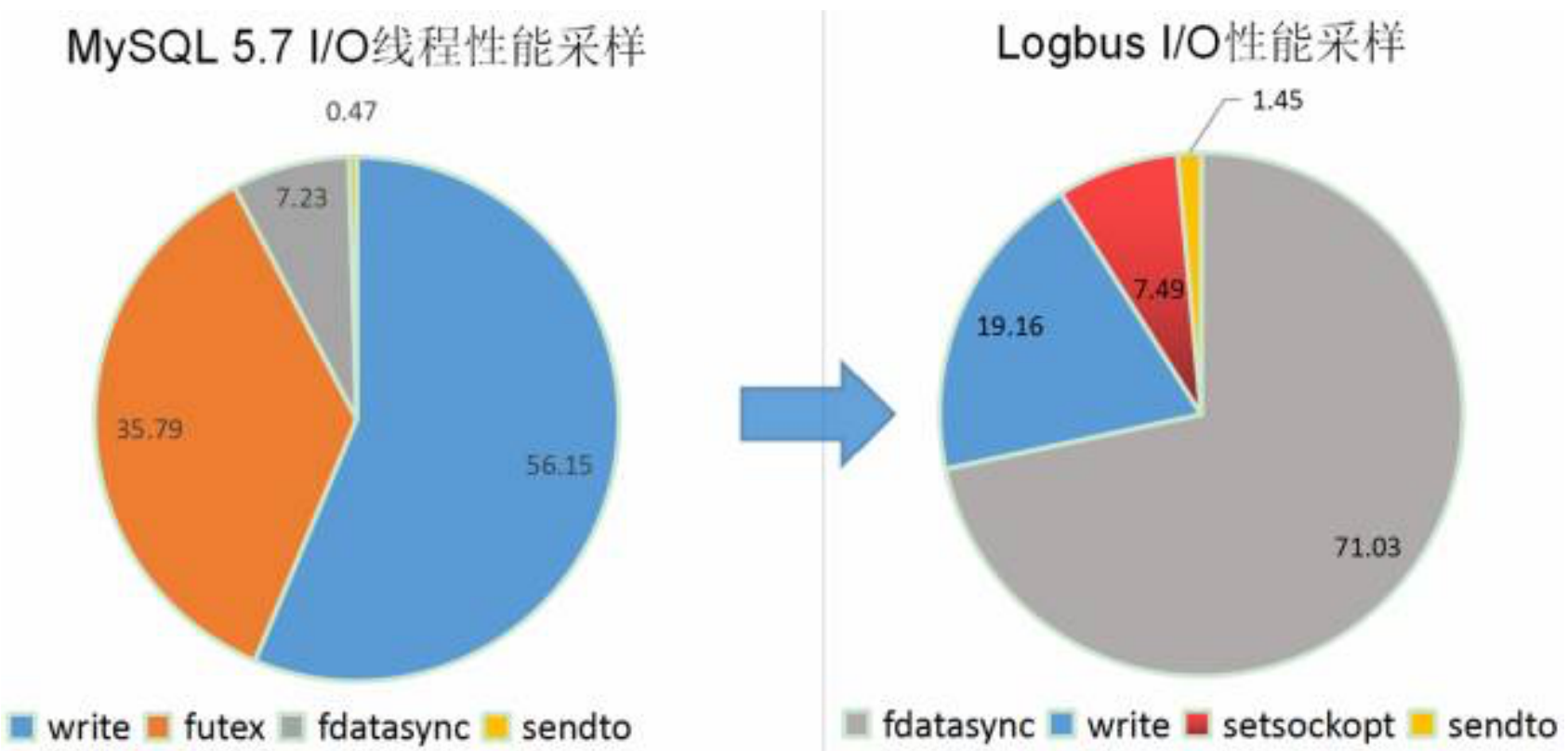
- 写入到Relay Log中，这个过程会和Slave SQL线程争抢保护Relay Log的锁。
- 更新复制元数据(包含Master的位置等信息)。

优化方法

经过分析，我们的优化策略是：

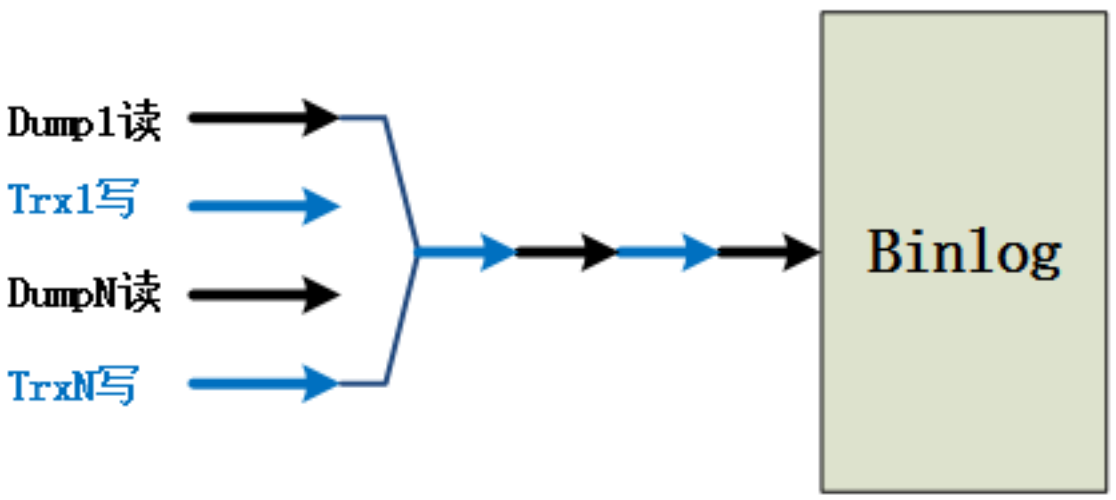
- Slave I/O线程和Slave SQL线程是典型的单写单读生产者-消费者模型，是可以做到无锁设计的；因此实现思路就是Slave I/O线程在每次写完数据后，原子更新Relay Log的长度信息，Slave SQL线程读取Relay Log的时以长度信息为边界。这样就将原本竞争激烈的Relay Log锁化解为无锁；
- 由于Binlog事件中的GTID(Global Transaction Identifier)和DB事务是一一对应的关系，所以Relay Log中的数据本身已经包含了所需要的复制元数据，所以我们可以不写Master info文件，消除了冗余的文件I/O；
- 于DB都是以事务为更新粒度的，因为在Relay Log文件I/O上，我们通过合并离散小I/O为事务粒度的大I/O等手段，使磁盘I/O得以大幅提升。

优化效果



如上图所示，经过优化：左图35.79%的锁竞争(futex)已经被完全消除；同压测压力下，56.15%的文件I/O开销被优化到19.16%，Slave I/O线程被优化为预期的I/O密集型线程。

2.主库事务线程和Dump线程间的优化



问题分析

如上图所示，在原生MySQL中多个事务提交线程TrxN和多个Dump线程之间会同时竞争Binlog文件资源的保护锁，多个事务提交线程对Binlog执行写入，多个Dump线程从Binlog文件读取数据并发送给Slave。所有的线程之间是串行执行的！

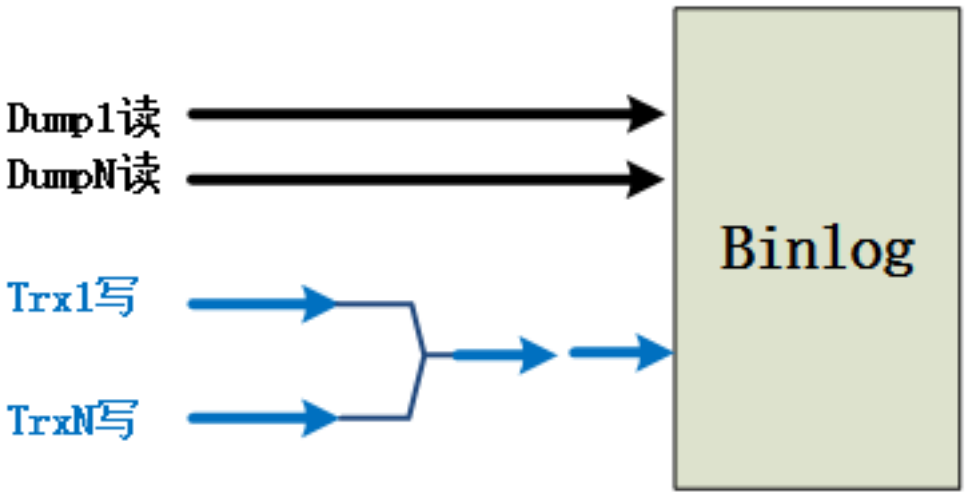
优化方法

经过分析，我们的优化策略是：

- 将读写分离开来，多个写入的线程还是在锁保护下串行执行，每一个写入线程写入完成后更新当前Binlog的长度信息，多个Dump线程以Binlog文件的长度信息为读取边界，多个Dump线程之间并行执行。以这种方式来让复制拓扑中的Dump线程发送得更快！

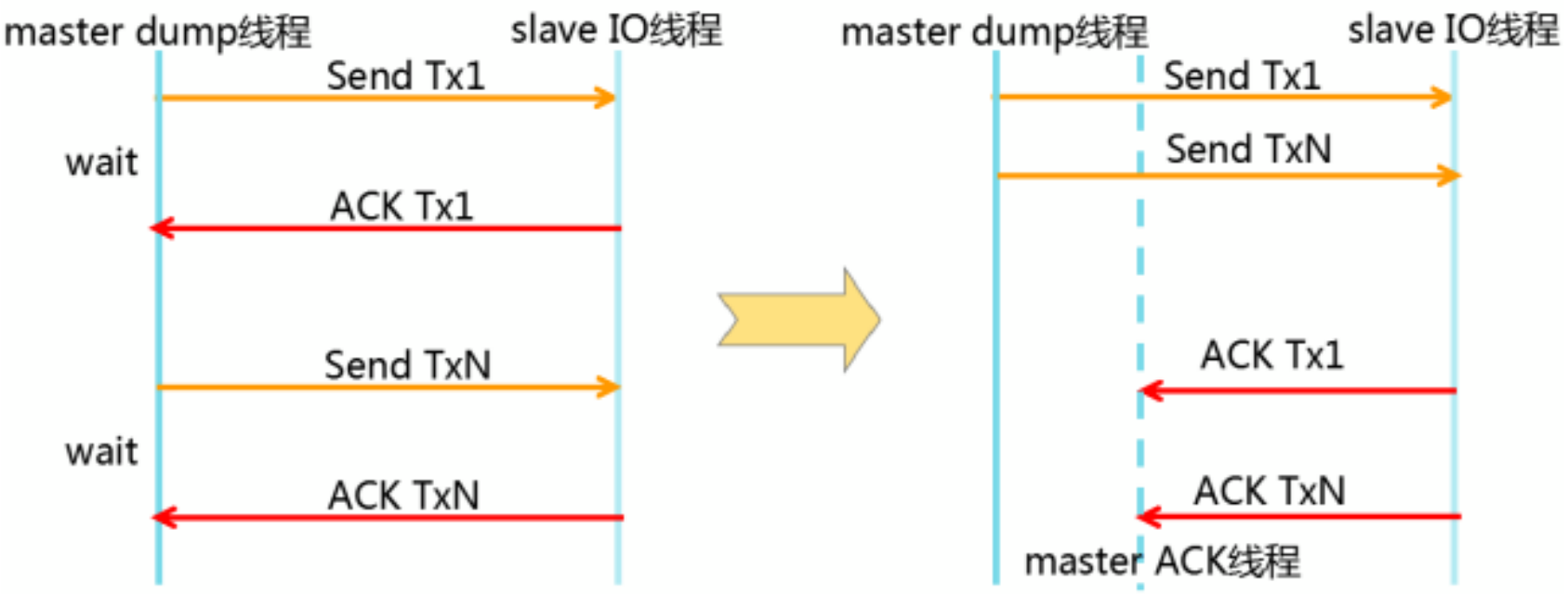
效果

优化后的示意图如下：



经过测试，优化后的内核，不仅提升了事务提交线程的性能，在Dump线程较多的情况下，对主从复制性能有较大提升。

二.主备库交互流程优化



问题分析

如上图所示，在原生MySQL中主备库之间的数据发送和ACK回应是简单的串行执行，在上一个事件ACK回应到达之前，不允许继续发送下一个事件；这个行为在跨园区(RTT 2-3ms)的情况性能非常差，而且也不能很好地利用带宽优势。

优化方法

经过分析，我们的优化策略是：

- 将发送和ACK回应的接收独立到不同的线程中，由于发送和接收都是基于TCP流的传输，所以时序性是有保障的；这样发送线程可以在未收ACK之前继续发送，接受线程收到ACK后唤醒等待的线程执行相应的

任务。

效果

根据实际用例测试，优化后的TPS提升为15%左右。

三.内核功能的优化

1. 预留运维帐号连接数配额

在腾讯云上，不时遇到用户APP异常或者BUG从而占满DB的最大连接限制，这是CDB OSS帐号无法登录以进行紧急的运维操作。针对这个现状，我们在MySQL内核单独开辟了一个可配置的连接数配额，即便在上述场景下，运维帐号仍然可以连接到DB进行紧急的运维操作。极大地降低了异常情况下DB无政府状态的风险。该帐号仅有数据库运维管理权限，无法获取用户数据，也保证了用户数据的安全性。

2. 主备强同步

针对一些应用对数据的一致性要求非常高，CDB在MySQL原生半同步的基础上进行了深度优化，确保一个事务在主库上提交之前一定已经复制到至少一个备库上。确保主库宕机时数据的一致性。

四.外围系统的优化

除了以上提到的MySQL内核侧的部分优化，我们也在外围OSS平台进行了多处优化。例如使用异步MySQL ping协议实现大量实例的监控、通过分布式技术来加固原有系统的HA/服务发现和自动扩容等功能、在数据安全/故障切换和快速恢复方面也进行了多处优化。