mysql的组提交

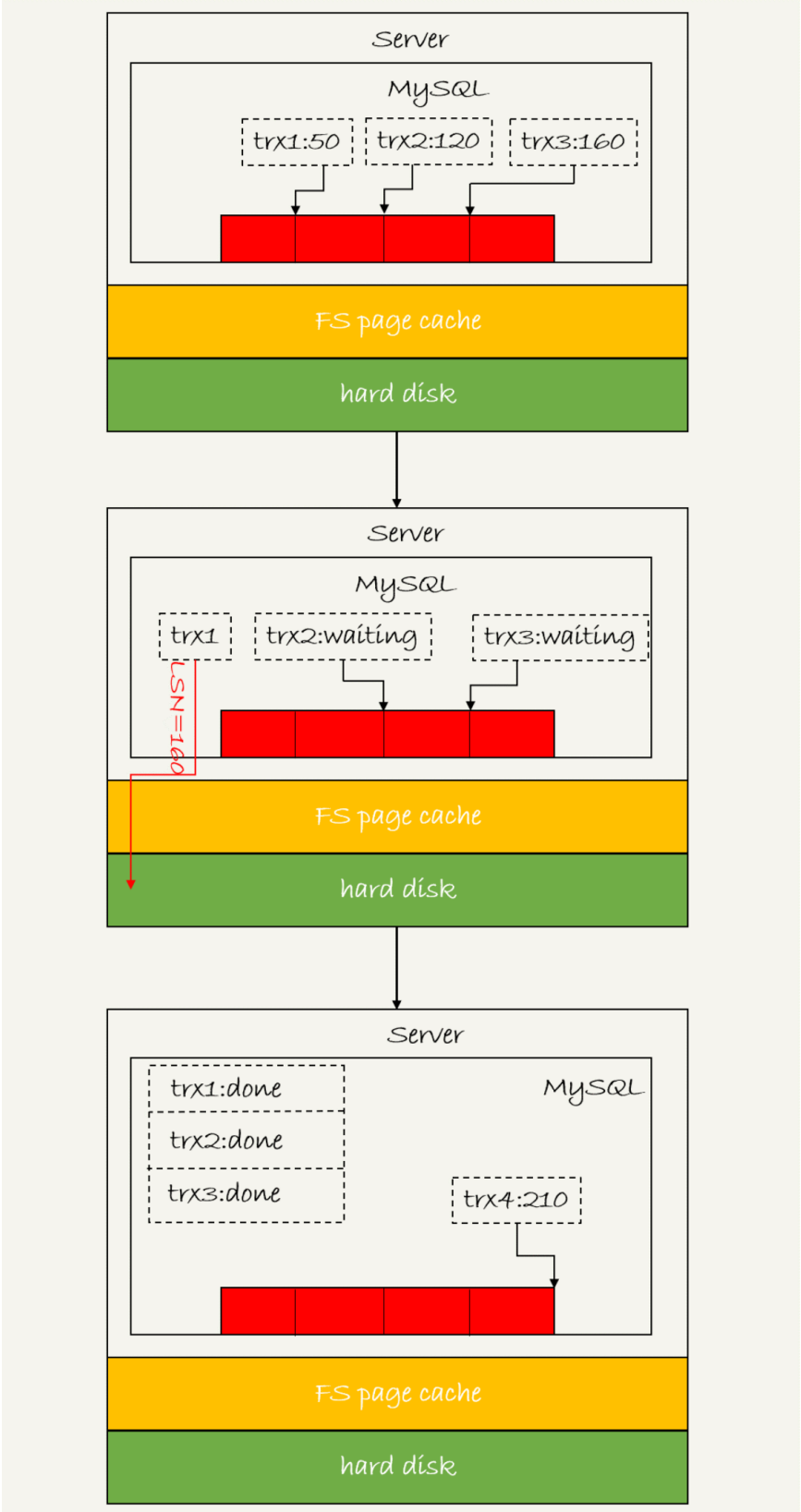
组提交（group commit）是mysql处理日志的一种优化方式，主要为了解决写日志时频繁刷磁盘的问题。组提交伴随着mysql的发展，已经支持了redo log和bin log的组提交。

1、redo log的组提交

WAL(Write-Ahead-Logging)是实现事务持久性的一个常用技术，基本原理是在提交事务时，为了避免磁盘页面的随机写，只需要保证事务的redo log写入磁盘即可，这样可以通过redo log的顺序写代替页面的随机写，并且可以保证事务的持久性，提高了数据库系统的性能。虽然WAL使用顺序写替代了随机写，但是，每次事务提交，仍然需要有一次日志刷盘动作，受限于磁盘IO，这个操作仍然是事务并发的瓶颈。

组提交思想是，将多个事务redo log的刷盘动作合并，减少磁盘顺序写。Innodb的日志系统里面，每条redo log都有一个LSN(Log Sequence Number)，LSN是单调递增的。每个事务执行更新操作都会包含一条或多条redo log，各个事务将日志拷贝到log_sys_buffer时(log_sys_buffer 通过log_mutex保护)，都会获取当前最大的LSN，因此可以保证不同事务的LSN不会重复。

假设现在有三个并发事务（tx1,tx2,tx3）,这三个事务所对应的LSN的值分别是50,120,160。


Server
MySQL
trx1:50
trx2:120
trx3:160
FS page cache
hard disk
Server
MySQL
trx1
trx2:waiting
trx3:waiting
LSN=160
FS page cache
hard disk
Server
MySQL
trx1:done
trx2:done
trx3:done
trx4:210
FS page cache
hard disk

从图中可以看到：

1、trx1是第一个到达的，会被选为这组的leader；

2、等trx1要开始写盘的时候，这个组里面已经有了三个事务，这时候LSN也变成了160,；

3、trx1去写盘的时候，带的就是LSN=160,因此等trx1返回时，所有LSN小于等于160的redo log都已经被持久化到磁盘；

4、这时候trx2和trx3就可以直接返回了

因此，在一个组提交里面，组员越多，节约磁盘IOPS的效果就越好。但如果只有单线程压测，那就只能老老实实地一个事务对应一次持久化操作了。

2、bin log的组提交

在之前的版本中，mysql的binlog是无法实现组提交的，原因在于redo log和binlog的刷盘串行化问题，而实现串行化的目的也是为了保证两份日志保持一致，而在5.6版本之后提供了一种解决方案，能够保证binlog实现组提交。基本思想是：引入队列机制保证 innodb commit顺序与binlog落盘顺序一致，并将事务分组，组内的binlog刷盘动作交给一个事务进行，实现组提交的目的。binlog提交将提交分为了3个阶段，flush阶段，sync阶段和commit阶段。每个阶段都有一个队列，每个队列有一个mutex保护，预定进入队列的第一个线程为leader，其他线程为follower，所有事情交给leader去做，leader做完所有的动作之后，通知follower刷盘结束。

具体流程可以看这篇帖子：

https://mp.weixin.qq.com/s/_LK8bdHPw9bZ9W1b3i5UZA

其实就是说如果想要提高binlog组提交的效率的话，那么可以通过设置一下两个参数：

binlog_group_commit_sync_delay 参数，表示延迟多少微秒后才调用fsync;

binlog_group_commit_sync_no_delay_count 参数，表示累积多少次以后才调用 fsync。