

MySQL XA事务存在的两个问题的复现

1 测试环境说明

序号	角色	版本
1	主库	Oracle MySQL 5.7.21-debug-log
2	从库	Oracle MySQL 5.7.21-debug-log

注：主从数据同步是基于binlog+pos的异步复制

2 场景

2.1 场景1: 主库存在悬挂事务

2.1.1 测试步骤

序号	步骤
1	主库执行： xa start 'xid1';
2	主库执行： insert into xa values (1, 1);
3	主库执行： xa end 'xid1';
4	主库执行： xa prepare 'xid1';
5	gdb调试主库的 mysqld , 在函数ha_commit_low 处断点
6	主库执行： xa commit 'xid1';, 此时触发断点
7	kill -9 MySQL (模拟crash)
8	重启MySQL主库
9	主库执行： xa recover; ,发现悬挂事务xid1
10	从库执行： xa recover; ,没有发现任何悬挂事务
11	主库执行： select * from xa where a=1; , 发现无此record
12	从库执行： select * from xa where a=1; , 发现有此record, 此时主从数据已不一致
13	主库执行： xa commit 'xid1';
14	主库执行： select * from xa where a=1; , 发现已有此record,主从数据恢复一致
15	从库执行： show slave status;查看主从复制状态, 发现主从复制中断, 报错： Error 'Unknown error 1397' on query. Default database: 'ght'. Query: 'XA COMMIT X'78696431',X',1'

2.1.2 结果分析

执行完第4步，主库和从库上的事务已经处于prepare状态。在执行第6步时，xa commit 已经通过复制发送到了从库，从库执行，完成该事务的commit。而由于主库上在引擎层提交的函数ha_commit_low处打断点，阻止了主库xa commit在各个引擎层的提交，因此主库上该事务没有提交成功，仍旧prepare状态。在第8步时，MySQL重启进入crash recovery阶段，InnoDB引擎层发现有处于prepare状态的外部XA事务，则将该事务插入到server层的xid_cache hash表中，xa recover命令可以查看xid_cache中所有的prepare阶段的外部XA事务。由于该XA事务在主库仍旧prepare阶段，尚未提交，而从库该事务已经提交，因此导致了12步中，主从不一致的现象。

而当在主库上执行第13步，将主库上该悬挂（处于prepare状态）事务提交时，该事务在主库commit，因此主从恢复数据一致性。

但是，由于从库上该事务已经提交，而非悬挂状态，因此13步的SQL复制到从库，执行时，找不到该悬挂事务，因此复制报错。

2.2 场景2: 从库存在悬挂事务

2.2.1 测试步骤

序号	步骤
1	主库执行：xa start 'xid2';
2	主库执行：insert into xa values (2, 1);
3	主库执行：xa end 'xid2';
4	gdb调试主库的mysqld，在函数trans_commit_stmt处断点
5	主库执行：xa prepare 'xid2';，此时触发断点
6	kill -9 MySQL (模拟crash)
7	重启MySQL主库
8	主库执行：xa recover;，发现无悬挂事务
9	从库执行：xa recover，发现有悬挂事务， 此时主从事务状态不一致
10	主库执行：select * from xa where a=2;，发现无此record
11	从库执行：select * from xa where a=1, 发现无此record, 数据是一致的，因为主从库上，该事务均未提交
12	从库执行：xa rollback 'xid2';， 此时主从库上事务状态和数据完全一致

2.2.2 结果分析

执行完步骤3，事务还没有进入prepare状态。在执行第5步时候，xa prepare已经通过复制发送到了从库，执行后，从库上该事务进入prepare状态。由于在函数trans_commit_stmt处打断点，阻止了主库上prepare的过程，prepare没有成功。在第7步时，MySQL重启进入

crash recovery阶段，主库上由于没有处于prepare状态的外部XA事务，因此不需要任何事务插入到xid_cache hash表中，xa recover命令查看不到任何处于悬挂（处于prepare状态）中的事务。而从库上的事务却已经prepare状态了，因此从库上通过命令xa recover是可以查看到存在悬挂事务的。

而当从库上执行第12步时，从库上的悬挂事务回滚，因此此时主从上事务状态和数据完全一致。