

KYLIN-5308 Epoch Renew 事务超时 Design

Root Cause

Epoch Renew 操作是为了当前Kylin节点更新自己所拥有的Epoch信息，也就是所谓的心跳。

Epoch Renew 只会更新当前节点拥有的Epoch

Renew 成功时，会在Epoch表的对应Epoch记录更新两个字段: last_epoch_renew_time、mvcc。

Renew 在更新数据库记录时，**有超时机制默认60s。超时后有重试机制。**

这里的问题就出在，**超时重试时，不会中断或者回滚旧事务对数据库的操作。**

并且更新数据库时，会检查Epoch记录的mvcc是否一致。

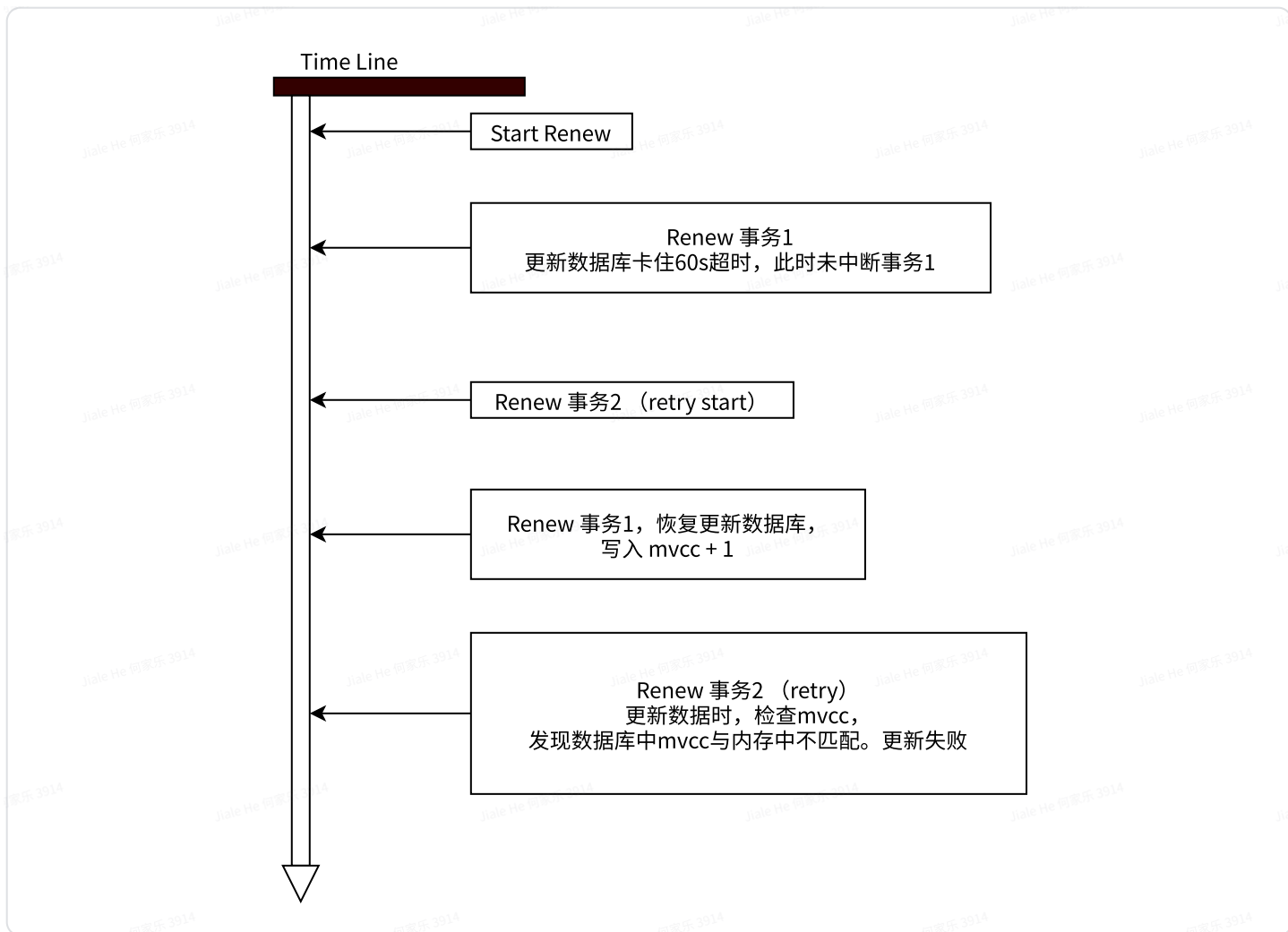
新事务还没写到数据库，旧事务完成了Epoch记录的更新使得mvcc + 1

这就导致了新事务在更新数据库时，发现mvcc不一致，查找不到要更新的数据记录，直接导致epoch renew失败，被kylin认为丢失了项目控制权。

```
org.apache.kylin.common.persistence.metadata.JdbcEpochStore#UPDATE_SQL
```

2 usages

```
static final String UPDATE_SQL = "update %s set " + EPOCH_ID + " =?, " + CURRENT_EPOCH_OWNER + " =?, " +  
    + LAST_EPOCH_RENEW_TIME + " =?, " + SERVER_MODE + " =?, " + MAINTENANCE_MODE_REASON + " =?, " + MVCC  
    + " =? where " + EPOCH_TARGET + " =? and " + MVCC + " =?";
```



Fix Design

原先Epoch超时时间由以下参数控制

```
kylin.server.leader-race.heart-beat-timeout=60s
```

1. kylin事务写数据库时, 新增事务超时回滚。
2. 新增配置 `kylin.server.leader-race.heart-beat-timeout-rate=0.8` , 用于控制事务超时时间, $60s \times 0.8 = 48s$ 事务超时时间。
3. 事务超时时, 事务的任何写操作会回滚, 不会影响别的事务。

UT

单元测试可以覆盖超时回滚。

主要在下面两个UT方法

- `JdbcEpochStoreTest#testExecuteWithTransactionTimeout_RollBack`
- `JdbcEpochStoreTest#testExecuteWithTransactionTimeoutSuccess`