mvcc正确性测试方案设计

我们使用对MemTable进行单测的方式，测试mvcc逻辑在大压力读写情况下的正确性，希望通过这个测试工具验证mvcc逻辑可能存在风险：

1. 事务数据提交后整体丢失；
2. 事务数据回滚不完全；
3. 事务隔离性被破坏：同一行数据同时被两个事务修改；
4. 事务原子性被破坏：事务提交后部分修改丢失；

设计两张表用于测试，名字分别为CheckInfo和CheckData。

CheckInfo表的schema如下：

Rowkey	RowStat	WriteThreadID	CheckDataValue

其中Rowkey为随机生成的整数；RowStat为int类型；WriteThreadID为int类型，保存写入这行的线程ID；CheckDataValue为varchar类型，保存待校验的数据，用于读取CheckData 表时进行校验。

CheckData表的schema如下：

Rowkey Column	SelfCheck Column1	SelfCheck Column2	MultiCheck Column1	MutiCheck Column2	MutiCheck ColumnN

其中Rowkey为自然数；SelfCheckColumn[1,2]用于行内的自校验，保存了最后一次修改这行的写线程ID；MultiCheckColumn[1...N]，用于与CheckInfo表的校验，不同的写线程修改不同的列，即MultiCheckColumn[N]只会被第N个写线程修改。

设计大小分别为N的写线程池和大小为M的读线程池，其中写线程用于生成随机事务写入MemTable，并产生一个校验任务推入队列(称为CheckQueue，使用现有的无锁队列ObFixedQueue)，读线程从任务队列中取出校验任务，根据任务描述读取MemTable进行数据校验。

写线程的循环处理流程如下：

1. 生成随机Rowkey(称为CI_RK)，用于写入CheckInfo表
2. 开启读写事务，读取CI_RK行，如果行存在，则判断RowStat是否为0，如果不为0则重新从1开始，否则继续下一步。后面的读写操作，如无特殊说明，均表示在本次事务内。
3. 将当前线程的ID，写入CheckInfo表的WriteThreadID列，将RowStat列的值改为1
4. 随机生成随机个数的连续自然数作为CheckData表的Rowkey(称为CD_RK)，这里允许多个写线程生成的CD_RK有重复
5. 随机生成数据，写入CheckData表中CD_RK行的MultiCheckColumn[N]列，并将当前线程ID写入SelfCheckColumn1列

6. 将第4，5步中生成的随机行列数据写入CheckInfo表的CheckDataValue列，用于读线程的校验

7. 开启只读事务，读取CheckData表中CD_RK行的其他MultiCheckColumn列

8. 将第7步读取到的数据，写回对应的行列，并计算整行的checksum写入SelfCheckColumn2列

9. 随机决定提交或回滚本次读写事务，如果事务提交，则将CI_RK推入CheckQueue

读线程的循环处理流程如下：

1. 从CheckQueue中读取到CI_RK

2. 根据CI_RK从CheckInfo表中读取WriteThreadID列和CheckDataValue列

3. 从CheckDataValue列的内容取出要读取CheckData表的rowkey，即CD_RK

4. 开启只读事务，根据第3步取出的CD_RK，分别构造Scan和MultiGet请求，查询CheckData表的SelfCheckColumn[1,2]列和MultiCheckColumn[N]列，并与从CheckInfo表中读取出的数据进行校验。

5. 修改CheckInfo中CI_RK行的RowStat为0

全局遍历校验：开启单独的线程池，运行如下检查：遍历CheckInfo表，对每一行执行上述2-4步。在内存写满后，停止写线程操作，执行一次全局遍历检查后，释放所有MemTable 后，重新开始新的一轮测试(这里不重新开始进程，而是在原来的进程内重复做)。在测试运行过程中，自动冻结活跃表，并在后续的读取中，自动合并冻结表和活跃表数据。