TDSQL全局一致性读技术详解

张文　数据库专家工程师

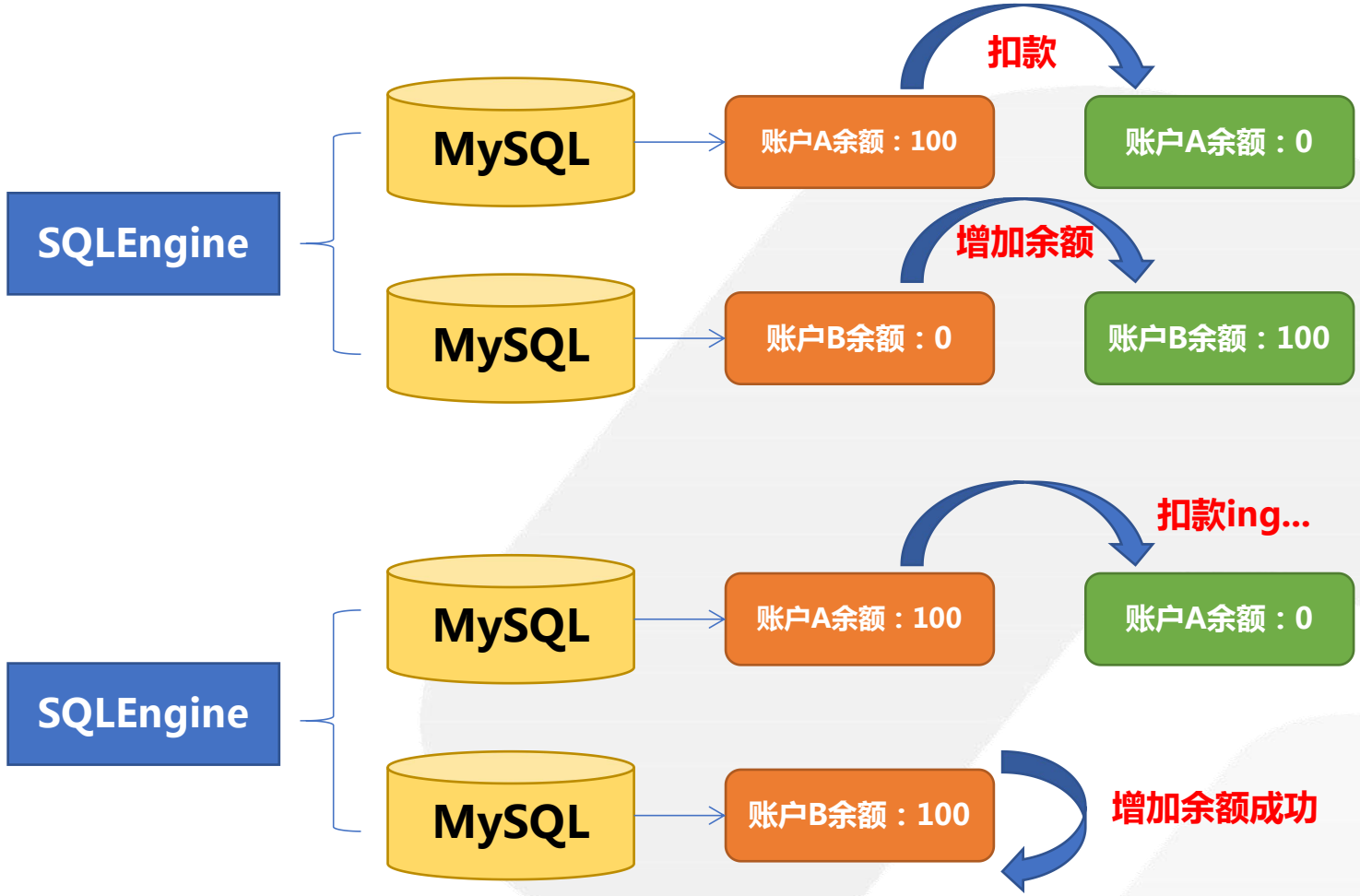
1、分布式下一致性读问题

2、TDSQL全局一致性读方案

3、一致性读下性能优化

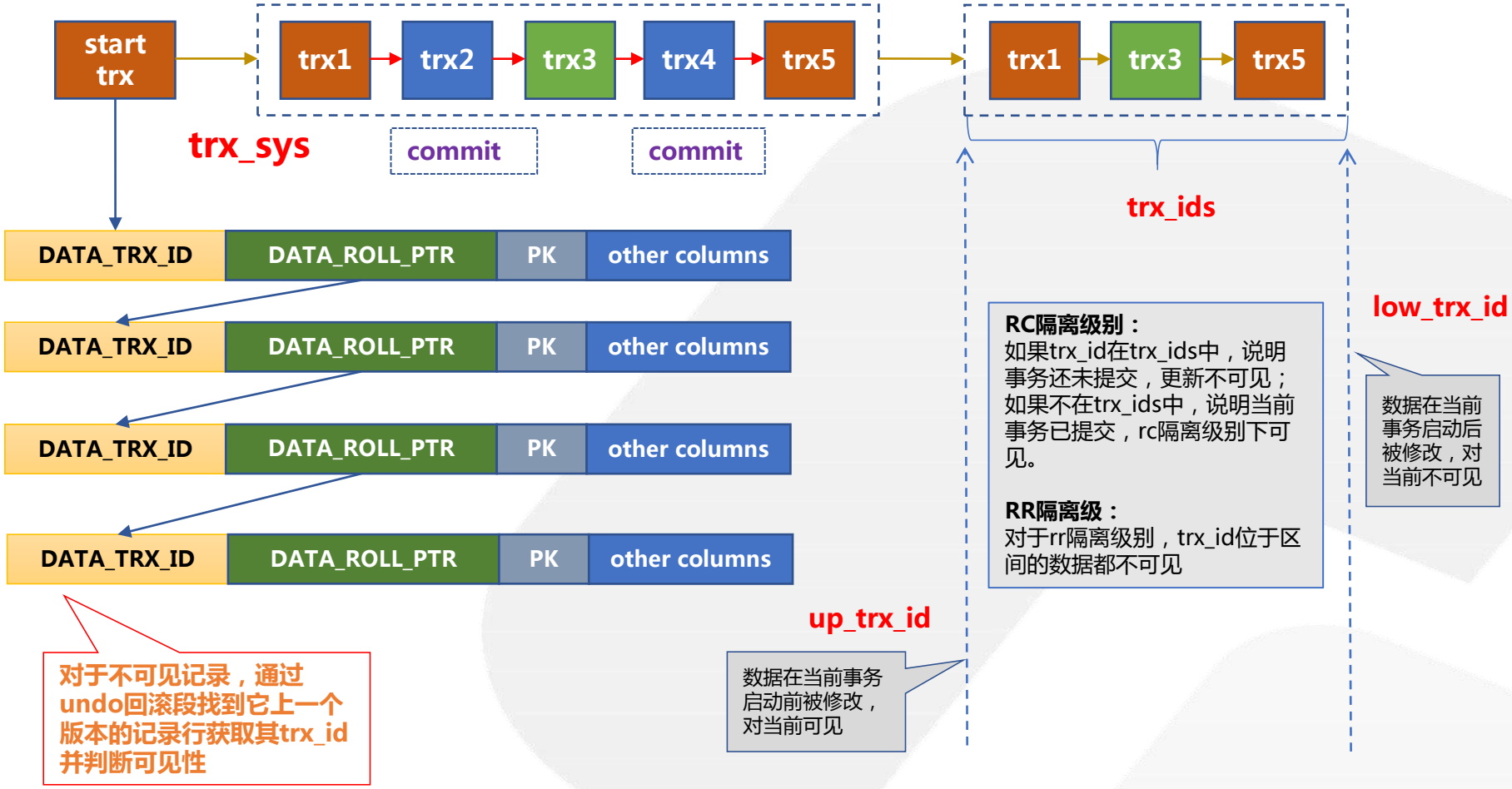
分布式场景下一致性读问题

- 分布式场景下不支持快照读，高并发场景下极易读取到分布式事务的“中间状态”
- 以转账交易为例，银行日结出报表的时候容易读取到“总账不平”的情况



存储节点的MVCC

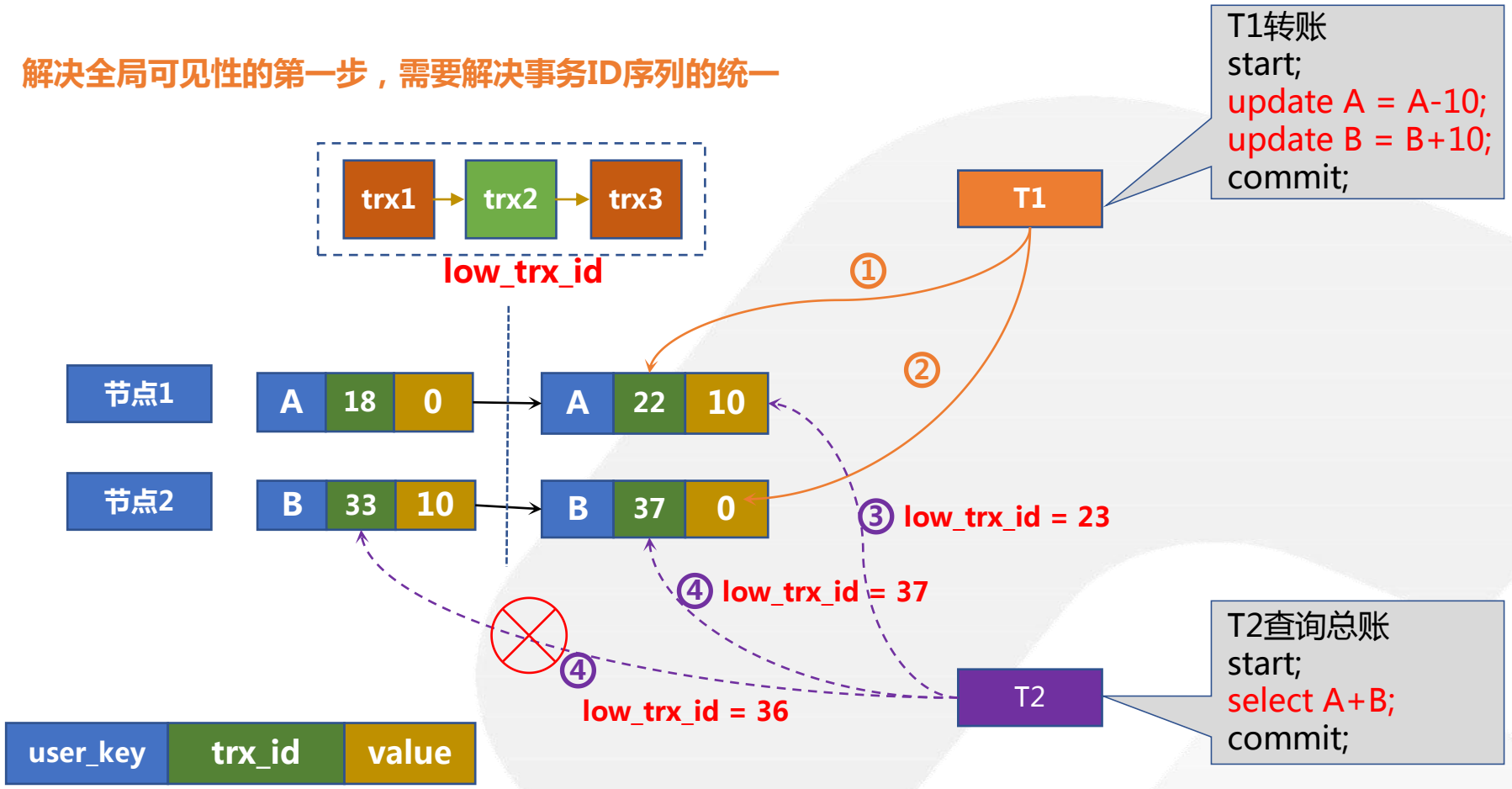
- 每个节点维护自身的MVCC可见性视图，节点之间互无关联
- 节点的可见性视图基于活跃事务链表以及事务的高低水位线实现



存储节点的MVCC

- 节点之间的事务ID序号彼此无关，各自有独立的生成规则
- 事务ID的不一致直接导致了可见性视图的不一致，最终导致这种中间状态

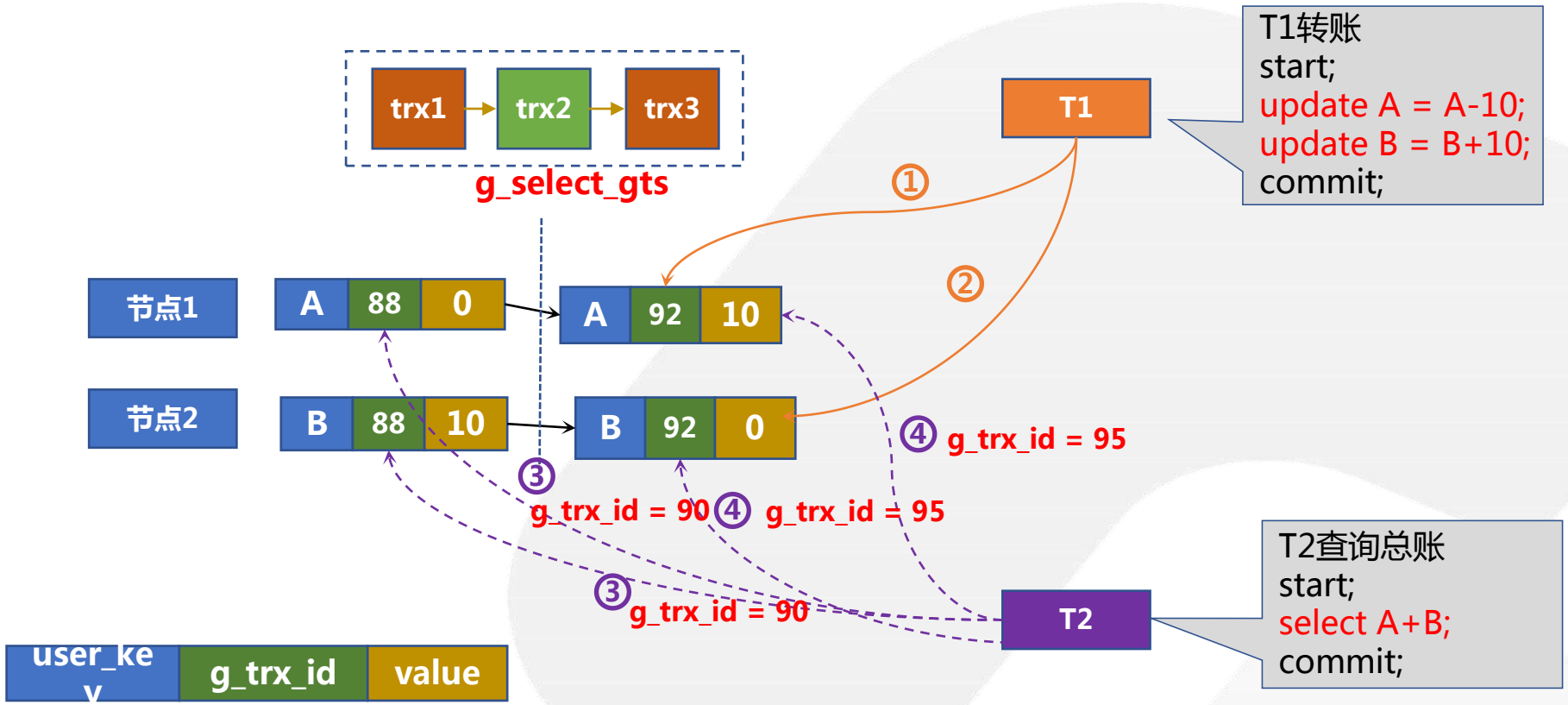
解决全局可见性的第一步，需要解决事务ID序列的统一



存储节点的MVCC

- 构建全局MVCC的前提：将事务ID转换为全局事务ID
- 每个节点基于全局事务ID，构建各自独立的可见性视图进而达到全局可见性

如何将建立事务ID和全局事务ID的映射是解决该问题的关键



TDSQL的新挑战

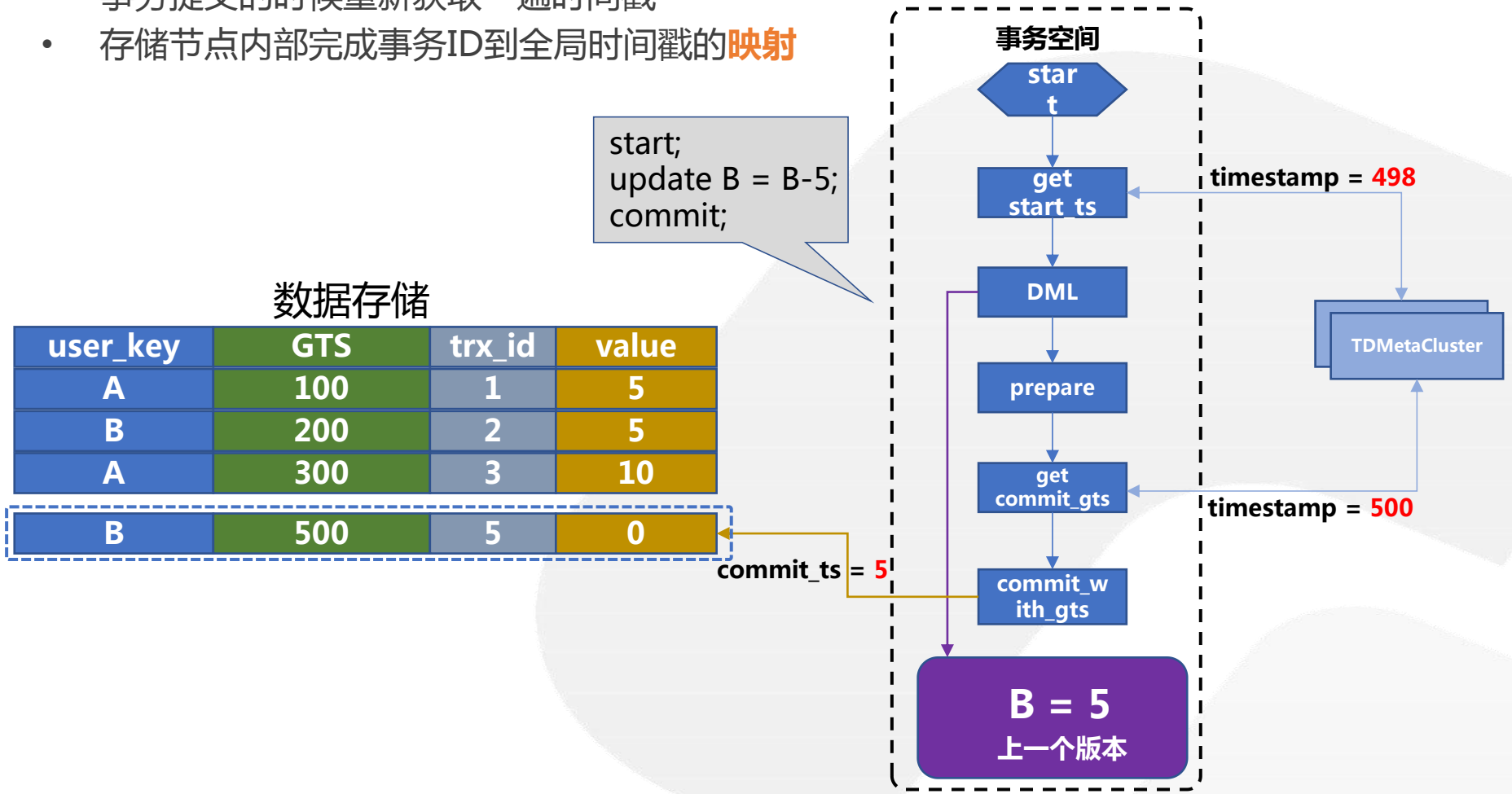
1、分布下式一致性读问题

2、TDSQL全局一致性读方案

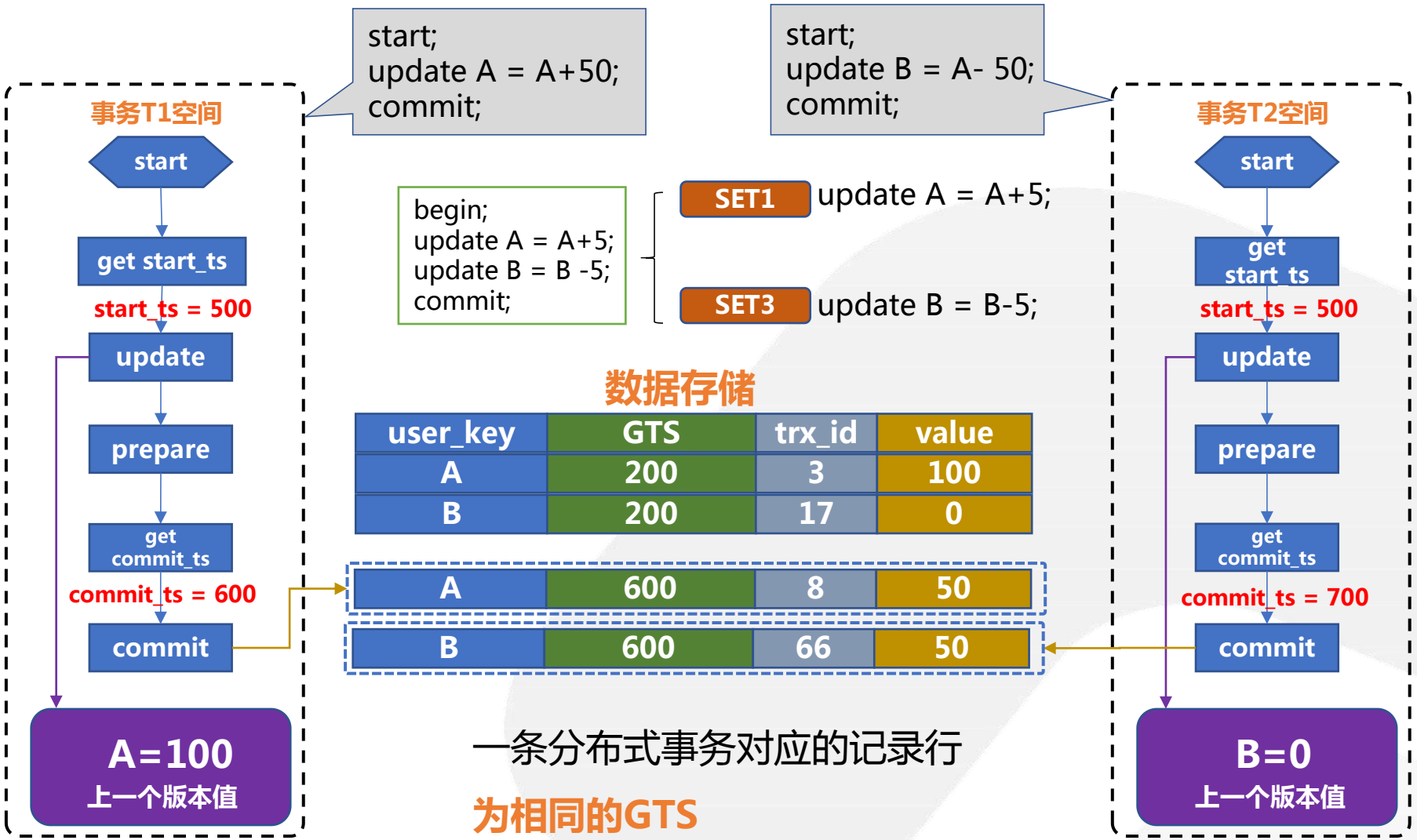
3、一致性读下性能优化

集群内引入全局时间戳

- 基于时间戳(timestamp)的数据**多版本**
- 全局时间戳服务(TDMetaCluster)，保证时间戳**全局单调递增**
- 开启事务时从MC获取当前的时间戳
- 事务提交的时候重新获取一遍时间戳
- 存储节点内部完成事务ID到全局时间戳的**映射**

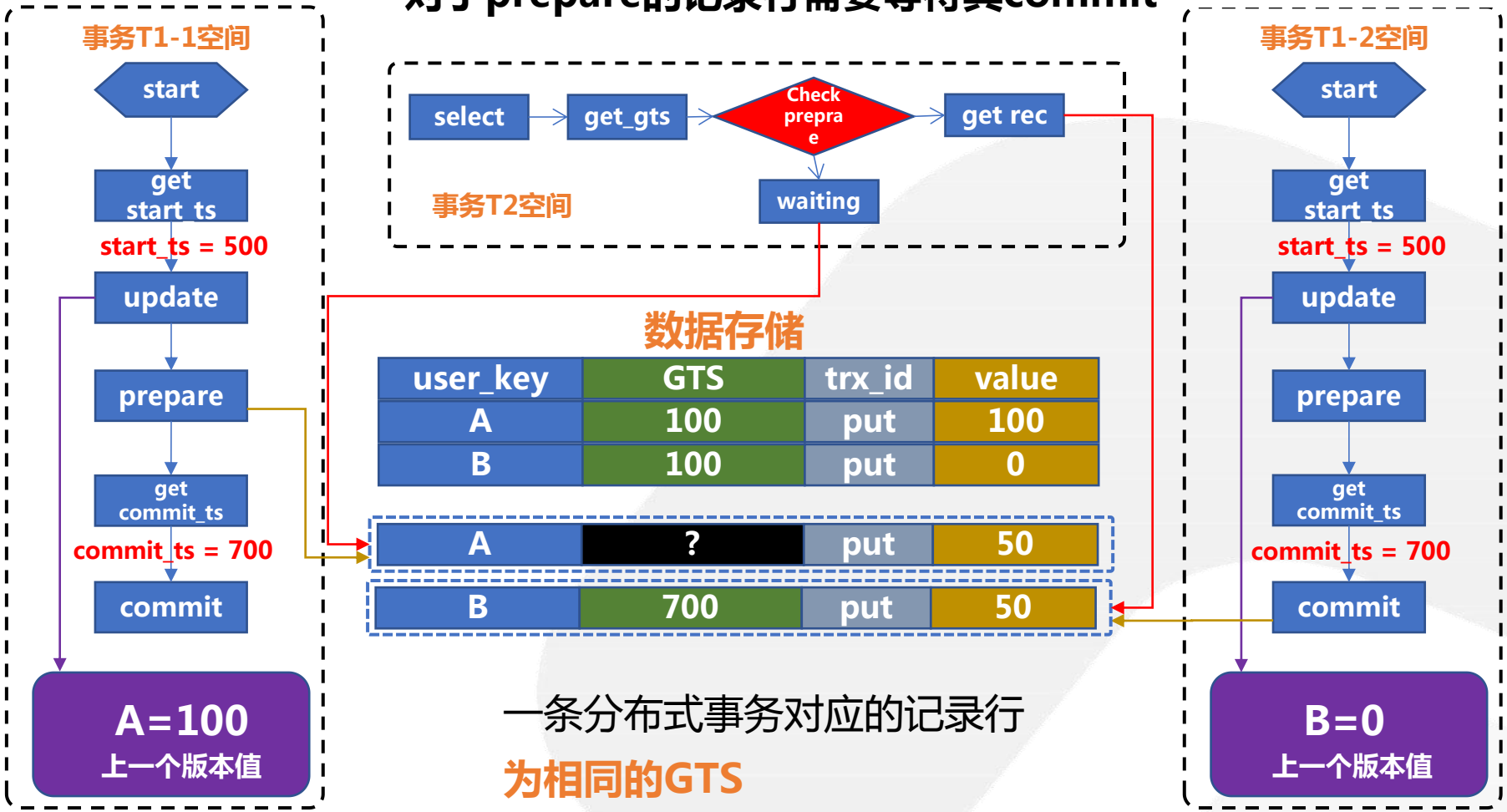


基于全局时间戳的一致性读



针对悬挂状态的事务

- 分布式事务两阶段提交prepare -> commit
- 对于prepare的记录行需要等待其commit



一条分布式事务对应的记录行

为相同的GTS

TDSQL的新挑战

1、分布式下一致性读问题

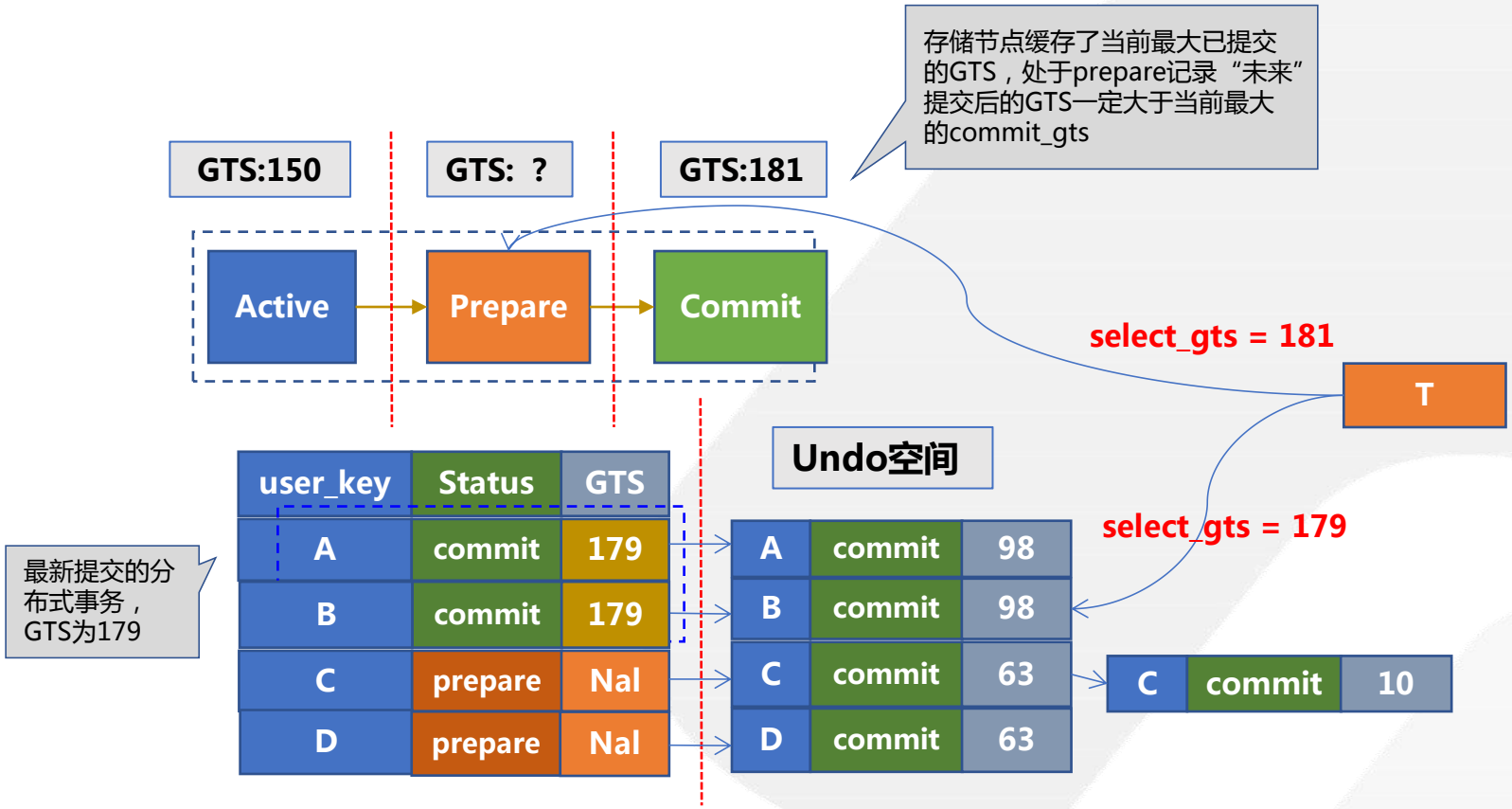
2、TDSQL全局一致性读方案

3、一致性读下性能优化

全局一致性读的性能瓶颈

全局一致性读性能瓶颈—-prepare等待问题

- 对于prepare记录需要做到尽可能**减少阻塞**select请求的场景
- 为了减少阻塞select的情况，必须为prepare状态的记录行**绑定全局时间戳**
- 为prepare记录行绑定时间戳的一种可行方式为**请求MC**
- 如果为prepare绑定时间戳，相当于每笔分布式事务需要**访问三次**MC
- 另外一种可行的方式为存储节点缓存一个**当前最大的commit** 时间戳



全局一致性读性能瓶颈—-非分布式事务问题

事务T1	事务T2
insert 交易日志	
	start GTS 100
	查询交易日志
	分布式事务
	commit
update 交易日志	

另一种可能的调度

事务T1	事务T2
	start GTS 100
	查询交易日志
	rollback
insert 交易日志	
update 交易日志	

业务连续性问题

另外一种带GTS的调度

事务T1	事务T2
	start GTS 100
	select A 0
start	
update A = 1	
get max_commit_gts	
	select A 0
commit gts 101	select A 0
	commit

事务T3提交后更新max_commit_gts为100，再递增+1为101

永远不可见

带GTS的调度

事务T1	事务T2
	start GTS 100
	select A 0
start	
update A = 1	select A 0
get max_commit_gts	
commit gts 100	select A 0
	select A 1
	commit

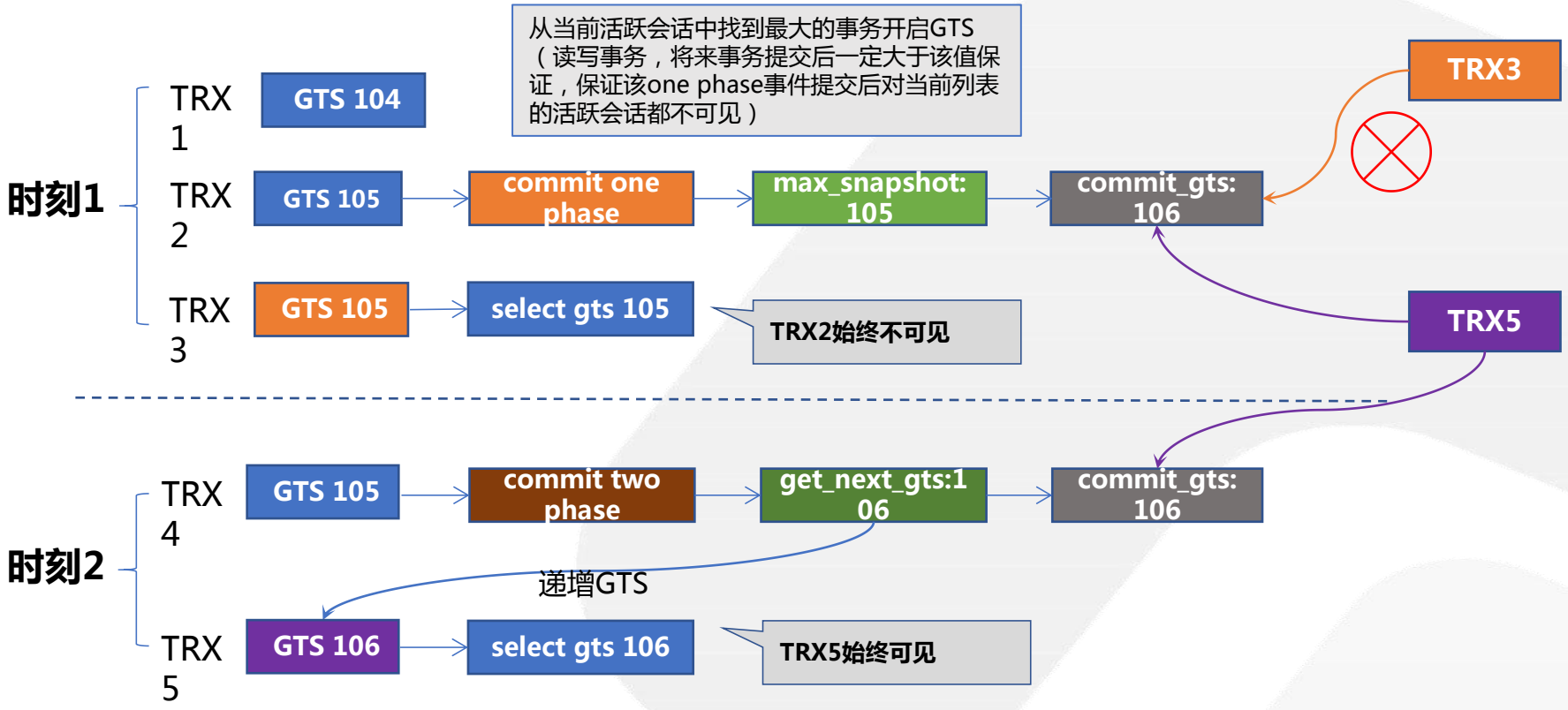
事务T3提交后更新max_commit_gts为100

不可重复读

全局一致性读性能瓶颈—-非分布式事务问题

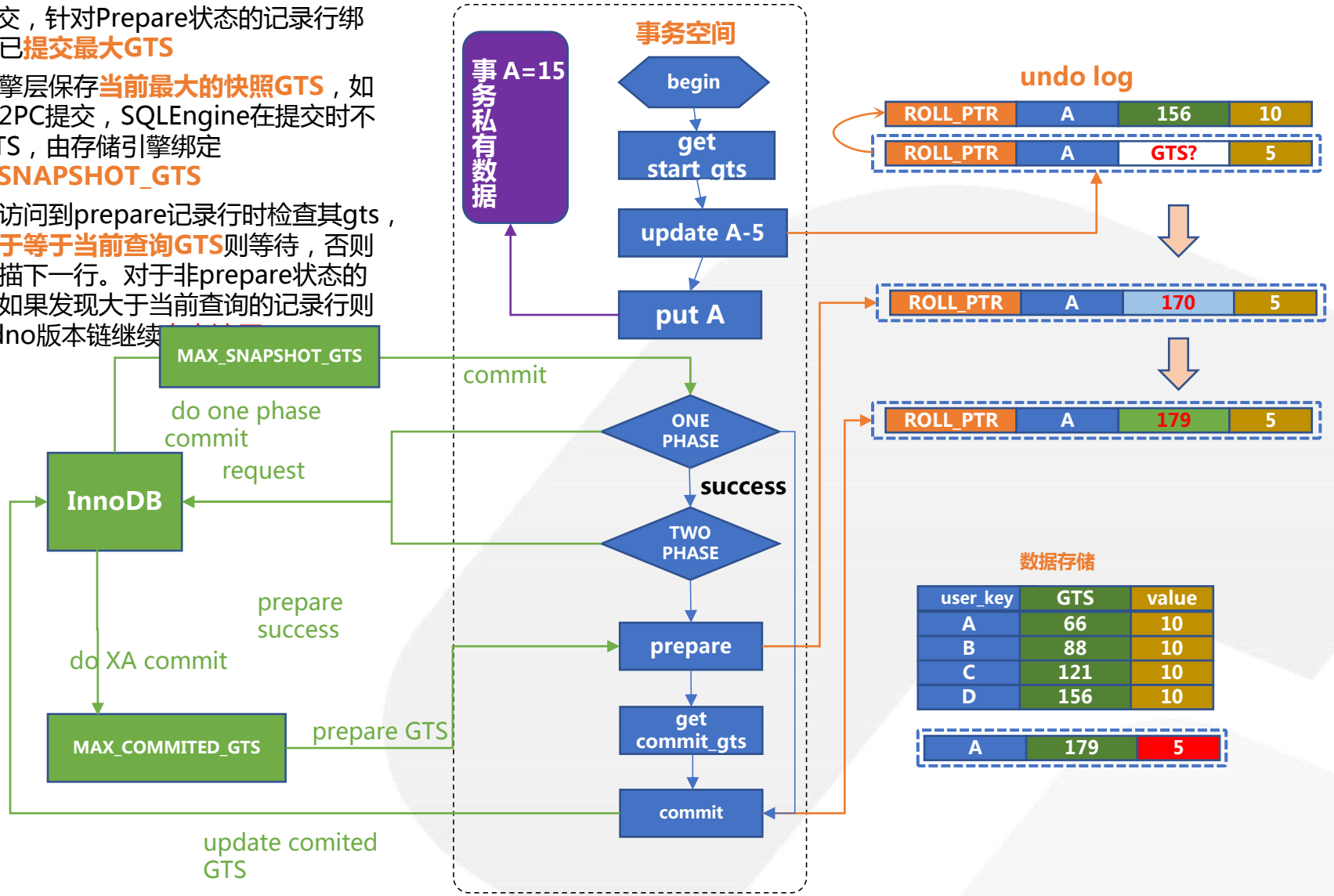
- 非分布式事务如果不绑定GTS会导致业务连续性受到影响
- 非分布式事务如果绑定max_commit_gts会导致不可重复读以及事务永远不可见问题

通过缓存活跃事务的快照GTS，来绑定一阶段提交的事务



全局一致性读性能瓶颈—-整体解决方案

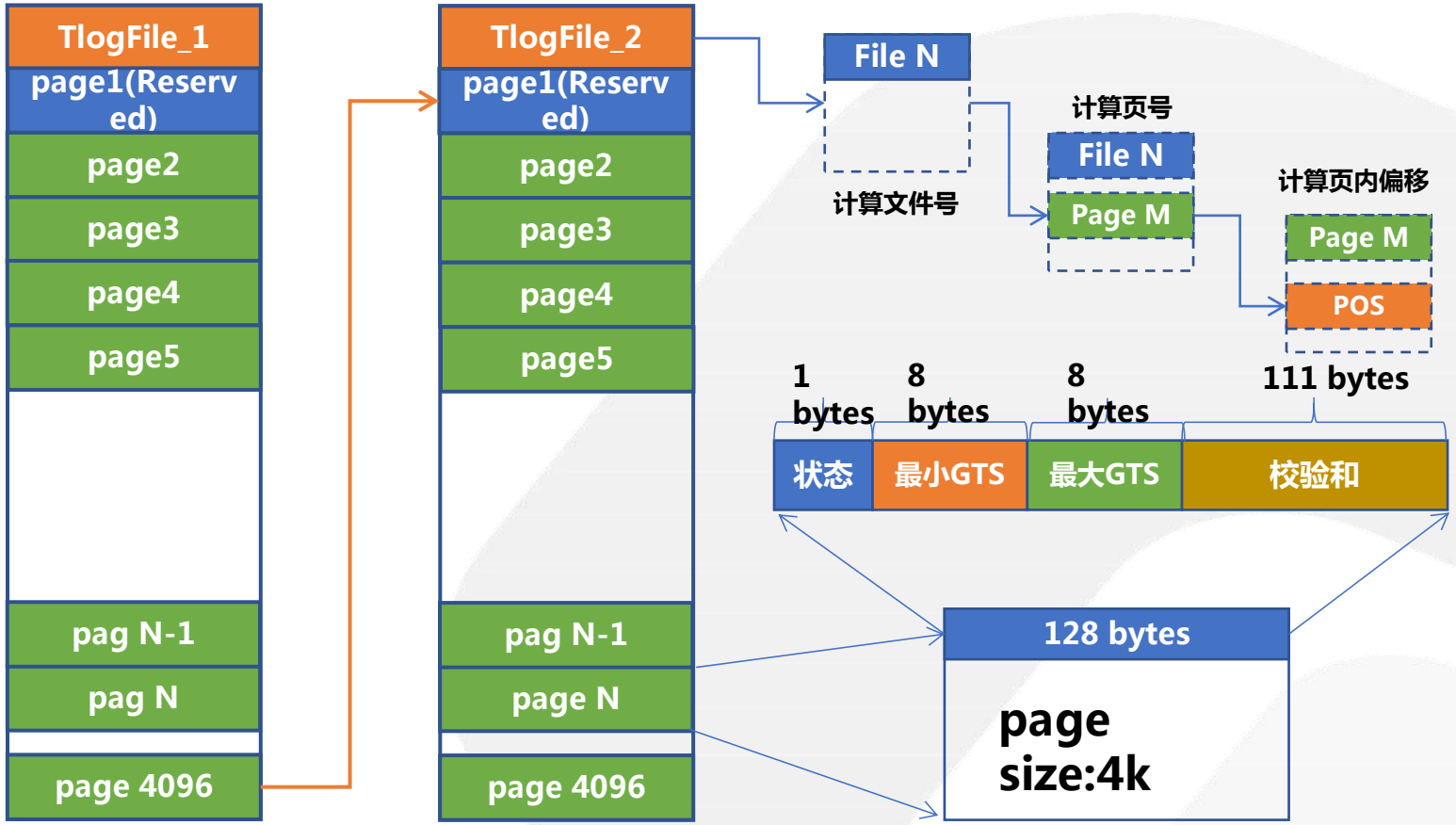
- 写事务在提交时进行2PC检测，如果走2PC提交，针对Prepare状态的记录行绑定当前已**提交最大GTS**
- 存储引擎层保存**当前最大的快照GTS**，如果是非2PC提交，SQLEngine在提交时不绑定GTS，由存储引擎绑定**MAX_SNAPSHOT_GTS**
- 读取事访问到prepare记录行时检查其gts，如果**小于等于当前查询GTS**则等待，否则继续扫描下一行。对于非prepare状态的记录行如果发现大于当前查询的记录行则通过udno版本链继续



一致性读技术挑战—-高性能映射

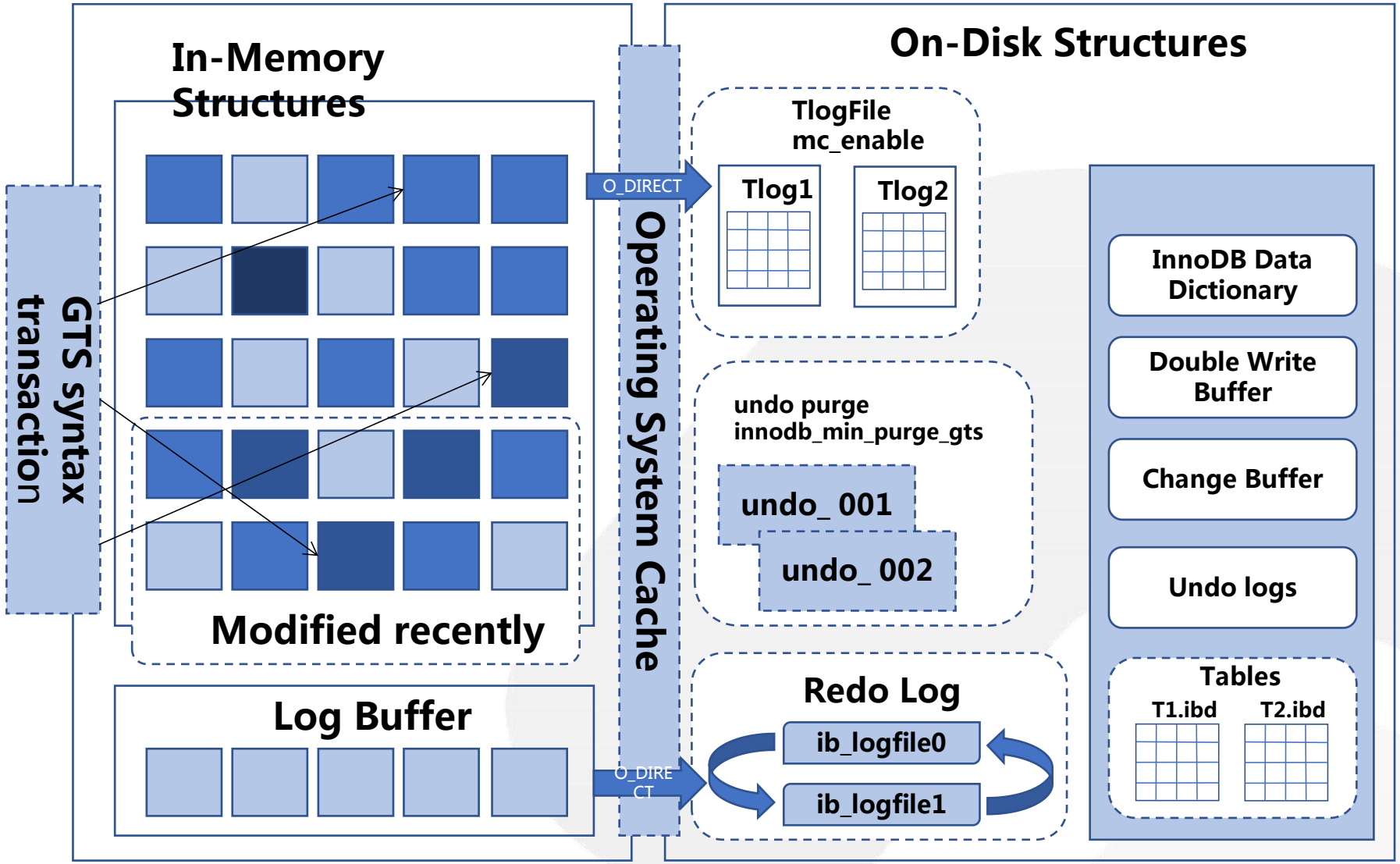
- GTS通过SQLEngine传入MySQL需要完成内部的映射
- 该映射关系需要做到高性能、持久化，MySQL意外Crash可以自动恢复

为了保证数据不丢不错需要先考虑其持久化，引入新的系统表空间Tlog



一致性读技术挑战—-高性能映射

如何利用MySQL本身的高性能架构，完成高效调度Tlog文件



扫码关注

“腾讯云数据库”

体验移动端运维数据库

获取更多资讯


THANKS
时序
开源
自研
分布式
文档
Spark
HANA
SQL
OLAP
Oracle
Hadoop
MPP
DB2