2019
05
08-10
北京新云南皇冠假日酒店

数据风云十年变迁

DTCC 第十届中国数据库技术大会

DATABASE TECHNOLOGY CONFERENCE CHINA 2019

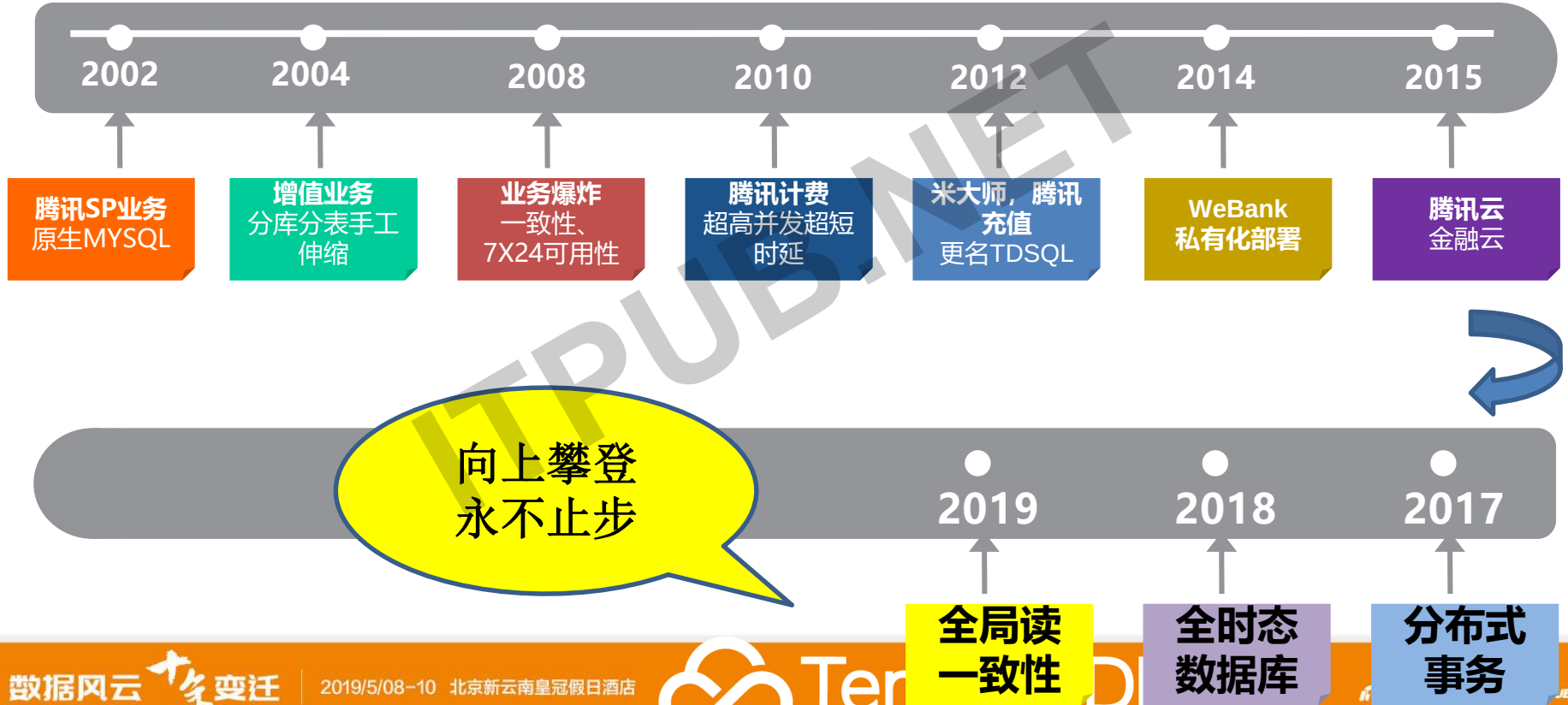
全局读一致性

---腾讯TDSQL分布式金融级数据库

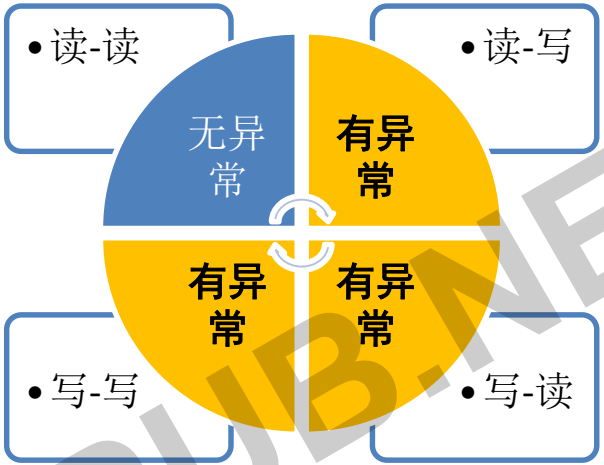
李海翔 @那海蓝蓝

面向金融类业务，十年积累，亿级账户验证

腾讯公司内与计费、充值、转账、财务等核心系统90%以上都使用TDSQL！



目录
CONTENTS



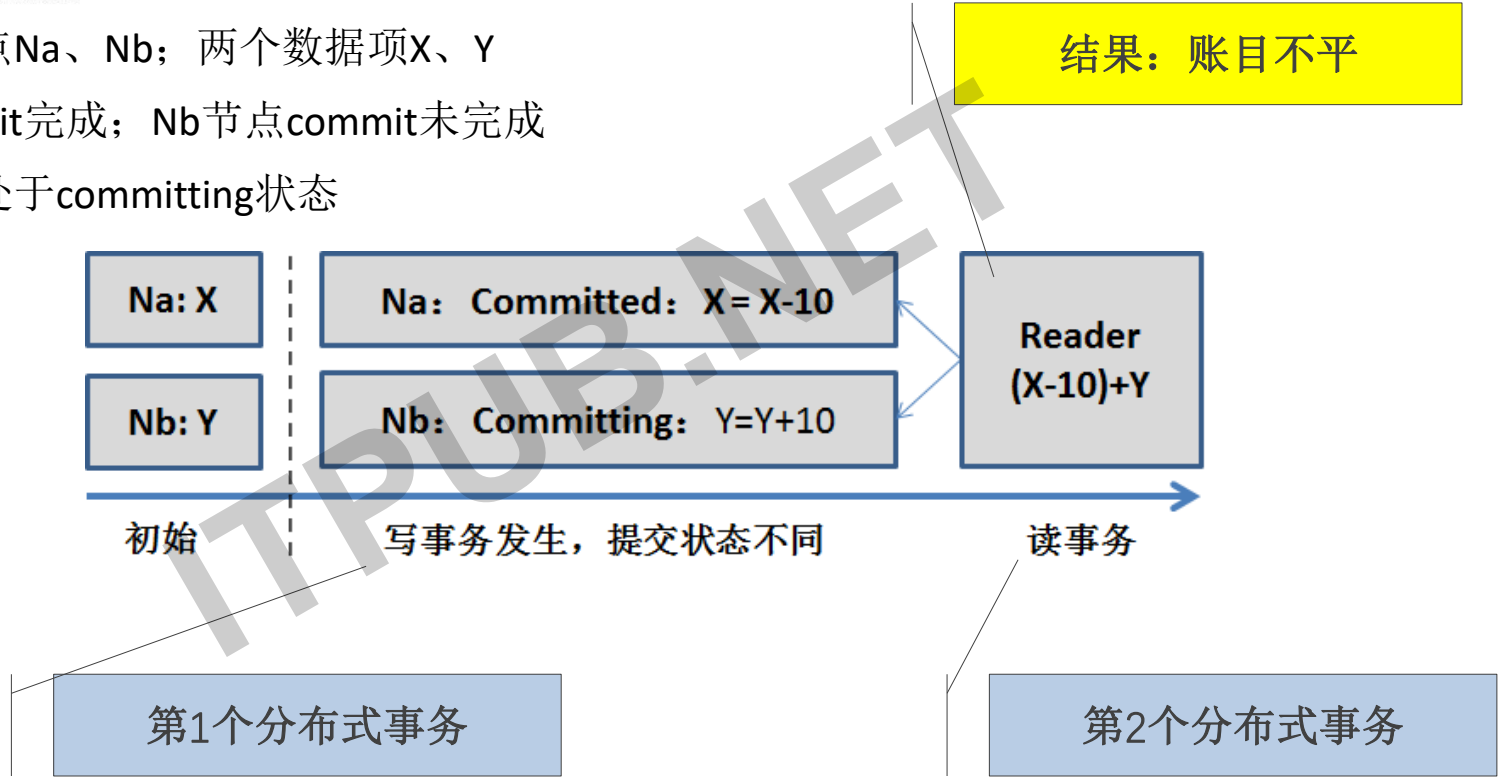
并发操作可以被区分为四种：读-读、读-写、写-读、写-写


各种数据异常
三种读异常
脏读
不可重复读
幻读
两种写异常
脏写
丢失更新
违反约束的异常（一致性）
违反语义的写偏序
其他异常
读偏序
游标丢失更新
Aborted Reads
Intermediate Reads


并发控制
主要技术
两阶段锁（2 Phase Lock）
时间戳（Timestamp Ordering）
基于有效性检查（validation protocal）
多版本和快照隔离（MVCC，Snapshot）
CO（Commitment Ordering）
OCC（Optimistic Concurrency Control）

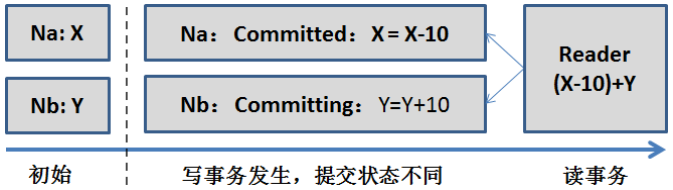
读半已提交数据异常

- 两个数据节点Na、Nb；两个数据项X、Y
- Na节点commit完成；Nb节点commit未完成
- 全局该事务处于committing状态



目录
CONTENTS

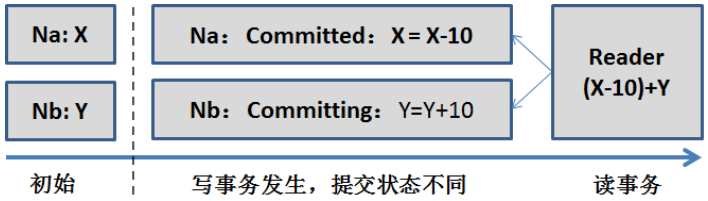
解决方案

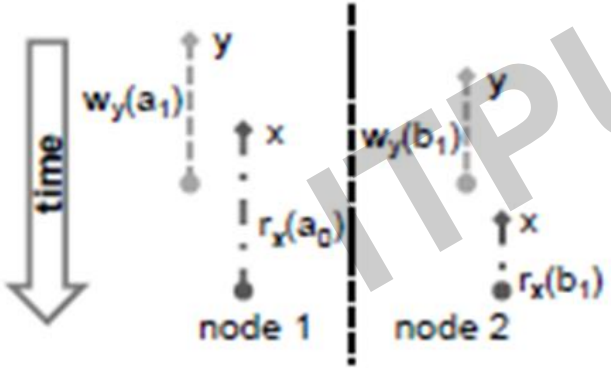


编号	各种方案	缺点	案例
1	全局事务管理器	非去中心化、低效	Pg XC
2	基于封锁的并发访问控制算法+全局可串行化	低效	某些系统 SS2PL+MVCC
3	全局可串行化+线性一致性	所有事件全序排序=>所有事务全局排序，低效	Spanner SS2PL+MVCC
4	全局可串行化+混合逻辑时钟+全局事务提交标志	数据是否可读，需要通过全局事务提交状态验证，增加通讯次数	CockroachDB SSI+MVCC
5	2次读 《Scalable atomic visibility with ramp transactions》	增加了通讯轮数，且只能解决读半已提交数据异常	学术界的解决方式

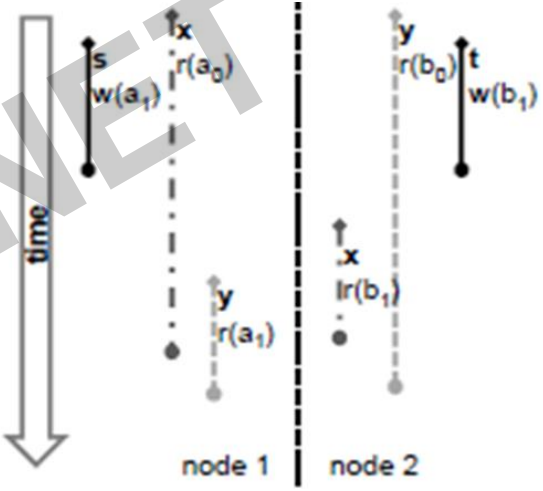
更多的数据异常.....

分布式读半已提交异常





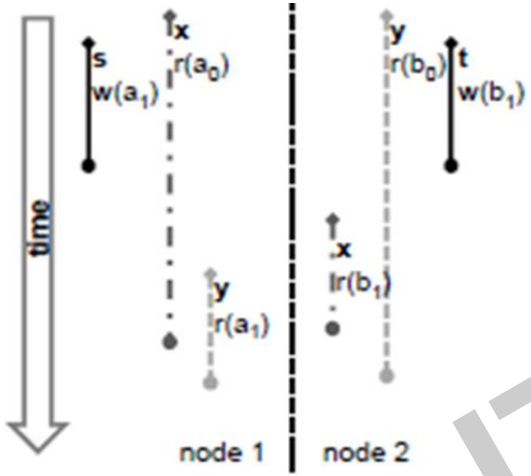
Cross异常

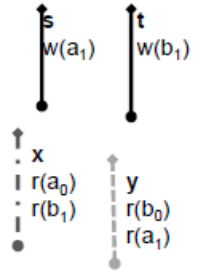


Carsten Binnig, Stefan Hildenbrand, Franz Färber, Donald Kossmann, Juchang Lee, Norman May: Distributed snapshot isolation: global transactions pay globally, local transactions pay locally. VLDB J. 23(6): 987-1011 (2014)

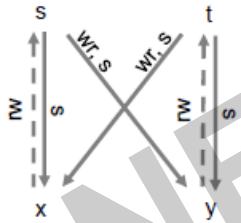
更多的数据异常.....

Cross异常

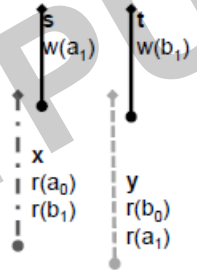




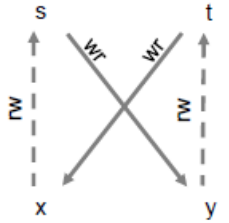
(b) Global schedule G1



(c) SSG(G1)



(d) Global schedule G2



(e) SSG(G2)

存在的业务问题：

- 事务x、y同时发起对账，对账结果不同

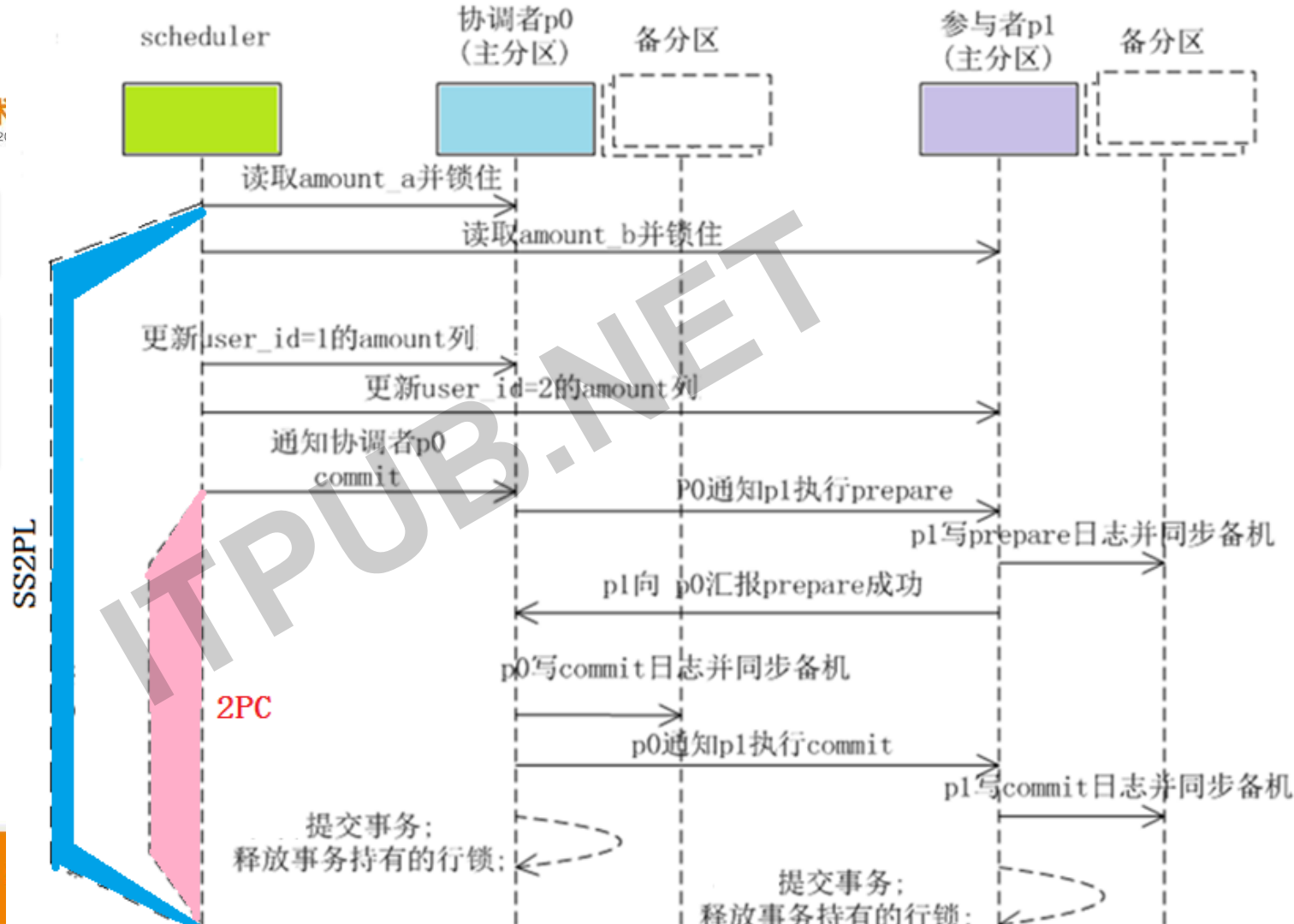
解决技术：

- 实现全局可串行化

目录
CONTENTS

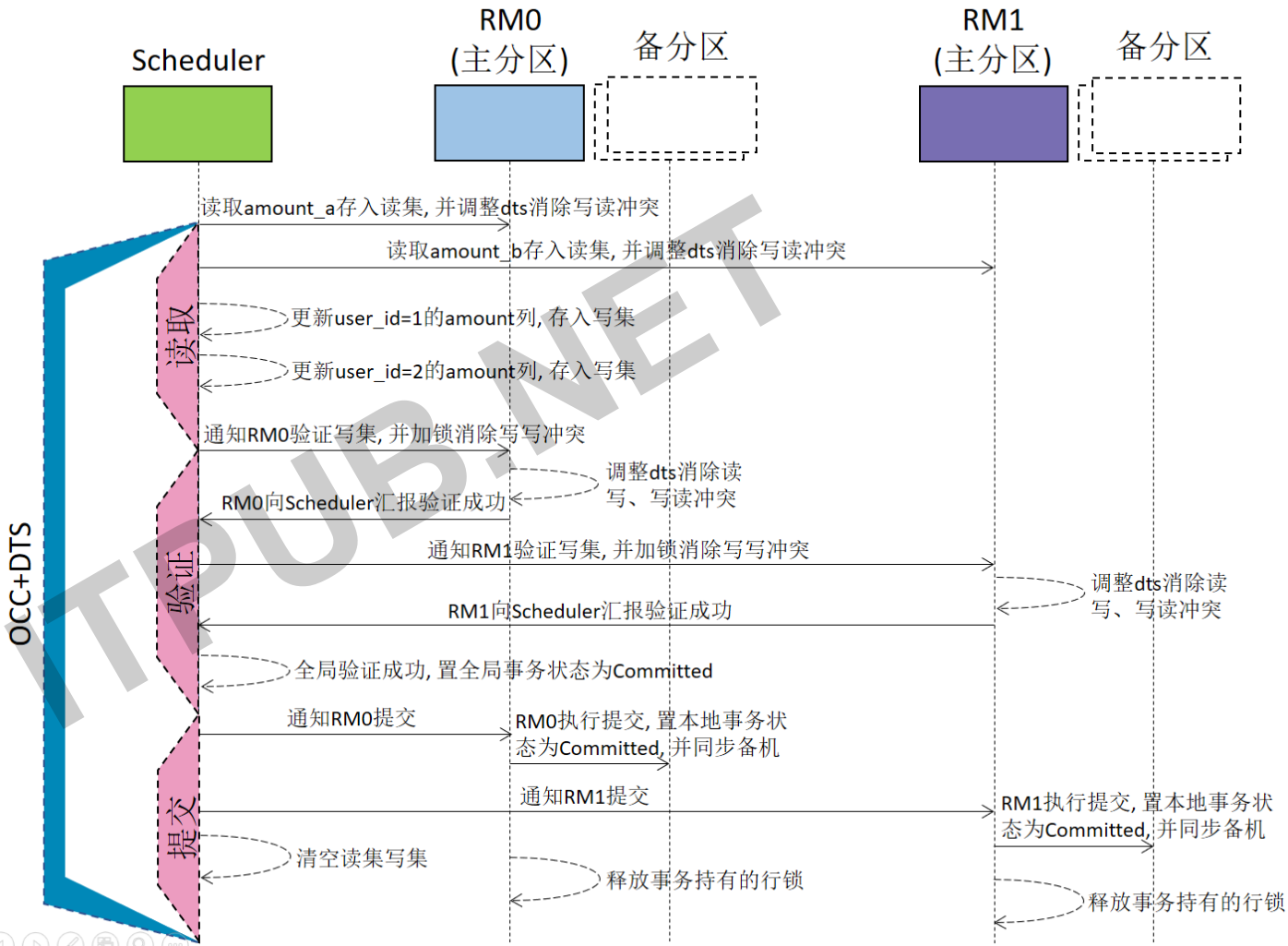
第一代TDSQL
分布式事务处理模型

start transaction;
修改user1的金额;
修改user2的金额;
commit;



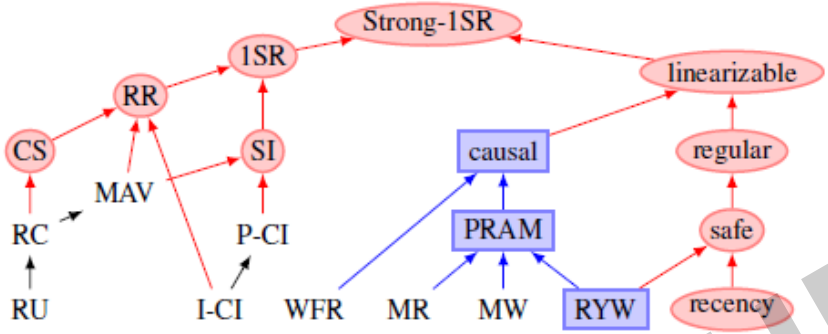
第二代TDSQL
分布式事务处理模型

start transaction;
修改user1的金额;
修改user2的金额;
commit;



全局一致性概念

Bailis P, Davidson A, Fekete A, et al. Highly available transactions: Virtues and limitations. In Proc. of VLDB. 2013. 181-192



Burckhardt [2014] breaks down linearizability into three components:

$$\text{LINEARIZABILITY}(\mathcal{F}) \triangleq \text{SINGLEORDER} \wedge \text{REALTIME} \wedge \text{RVAL}(\mathcal{F}). \quad (7)$$

where

$$\text{SINGLEORDER} \triangleq \exists H' \subseteq \{op \in H : op.oval = \nabla\} : vis = ar \setminus (H' \times H) \quad (8)$$

and

$$\text{REALTIME} \triangleq rb \subset ar. \quad (9)$$

and

$$\text{RVAL}(\mathcal{F}) \triangleq \forall op \in H : op.oval \in \mathcal{F}(op, cxt(A, op)). \quad (4)$$

In other words, SINGLEORDER imposes a single global order that defines both vis and ar, whereas REALTIME constrains arbitration (ar) to comply to the returns-before partial ordering (rb). Finally, RVAL($\mathcal{F}$) specifies the return value consistency of a replicated data type. We recall that, as per Equation (5), in case of read/write storage, this is the value written by the last write (according to ar) visible to a given read operation rd.

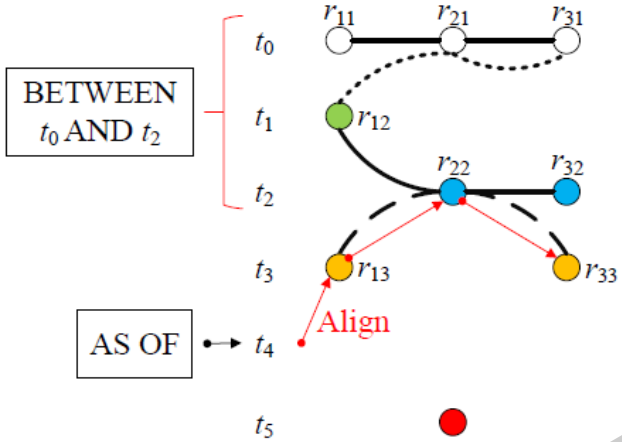
核心：同时满足2个一致

- 读的角度：分布式读数据一致性
- 读写混合的角度：分布式事务的事务一致性
- 非多副本的数据一致性

技术难点：

- 正确性相对容易实现
- 性能难以提高

解决方案—TDSQL全时态数据库



N1子节点	N2子节点	全局状态	是否可见
Prepared	Prepared	Preparing	不可见，读前一个版本
Prepared	Prepared	Prepared	不可见
Prepared	Prepared	Committed	可见
Committed	Prepared	Committed	可见
Committed	Committed	Committed	可见

核心问题：

- 分布式、全态数据在任何时间点的数据一致性

解决技术：

- 写写冲突封锁机制互斥
- MVCC从新版本到旧版本
- 局部节点处于Prepared状态
- 全局事务Committed/ Prepared状态
- 异步、批量设置本地事务状态
- 全局逻辑时钟（非跨城/洲分布）
- 冲突可串行化

- ***VLDB 2019 腾讯全时态论文《A Lightweight and Efficient Temporal Database Management System in TDSQL》***

未来展望