

# 全局一致性

OceanBase数据一致性的新飞跃！

---

乔国治（鸢腾）



OceanBase

# 目录

1. 故事的起源
2. 分布式架构面临的挑战
3. 业内常见解决方案
4. OceanBase的全局一致性

# 01

## 故事的起源

Snapshot Isolation & MVCC



# 如何“同时”读写同一条数据？

经典模式：

利用“**锁**”来实现互斥；“读”锁和“写”锁互斥，**先到先得，完全串行化**。

优点：

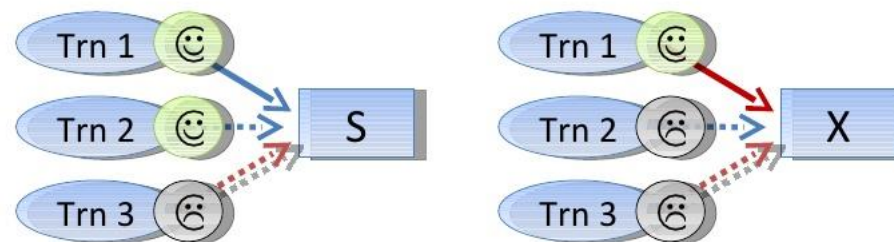
隔离性好，很容易利用不同类型的锁实现四种隔离级别（Serializable，Repeatable Read，Read Committed，Read Uncommitted）。

缺点：

读写互相影响，**并发性差**！

## Shared/Exclusive Lock

- S: shared lock for read
- X: exclusive lock for write



# Snapshot Isolation & MVCC

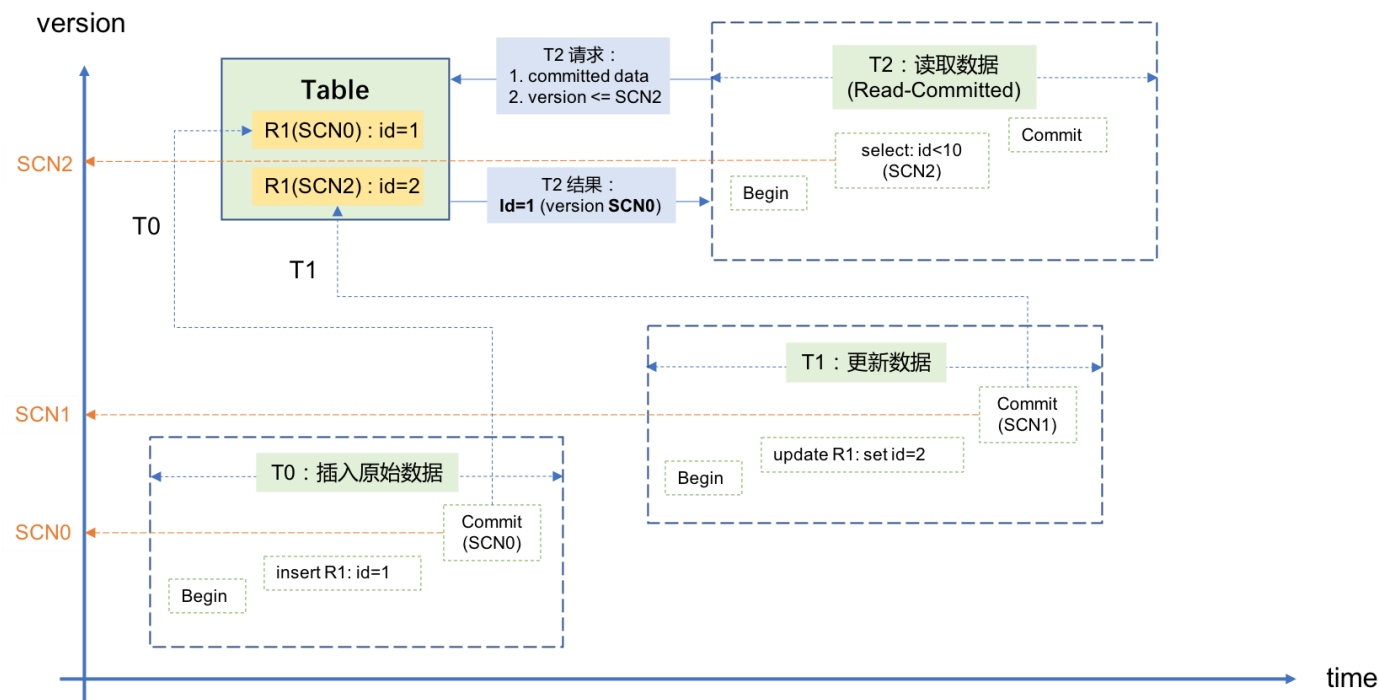
为数据维护多个版本（快照）。

**读操作只访问已提交的版本，不影响正在修改的版本。**

实现了对同一条数据的读写并发读写操作。

对数据隔离性几乎没有影响。

被众多数据库产品（如 Oracle、SQL Server、MySQL、PostgreSQL）广泛采用。OceanBase 也采用了这两种技术。



02

# 分布式架构面临的挑战

全局数据版本一致



# 多机器协同工作带来新问题

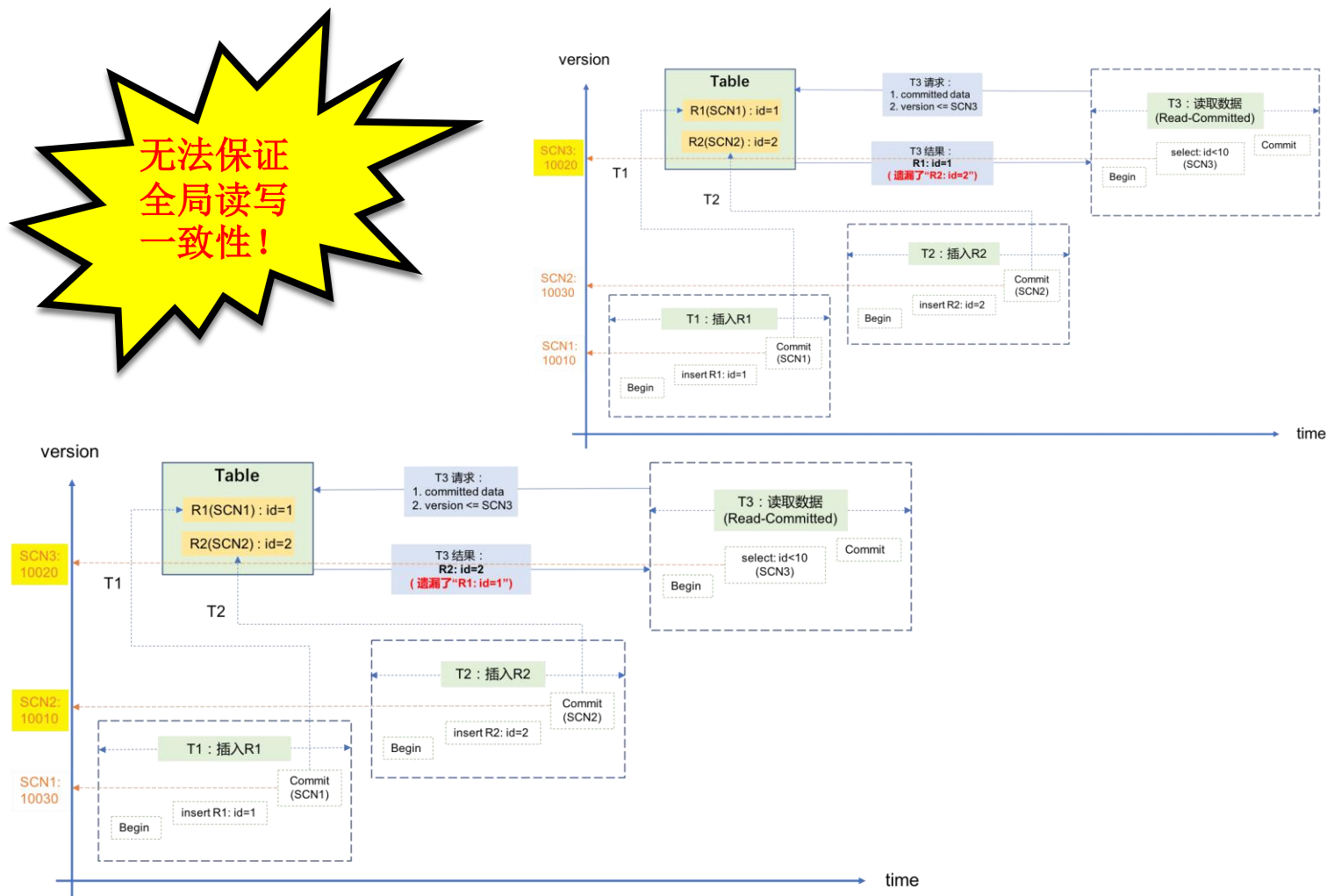
数据和事务涉及多台机器。

机器间有时钟差异。

继续使用本地时钟 ???

有问题。。。

无法保证  
全局读写  
一致性!



# 03 | 业内常见解决方案



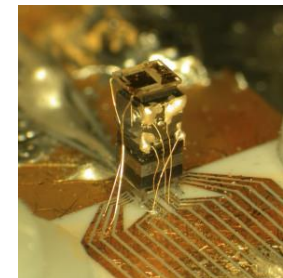


# 如何让版本号保持全局一致？

1. 高端硬件保证全局时钟同步。

软件实现简单。

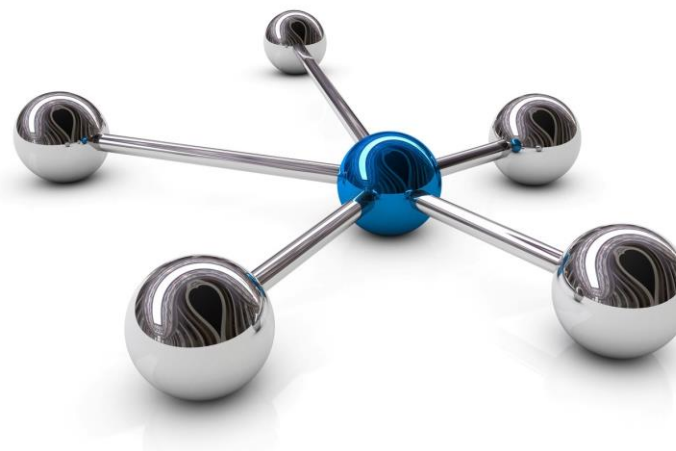
避免集中式服务的性能瓶颈。



2. 集中式的“全局版本号”服务。

解除对特殊硬件的强依赖。

彻底消除版本号乱序的可能。



# 04

# OceanBase的全局一致性

GTS & 全局一致性快照



# GTS ( Global Timestamp Service )

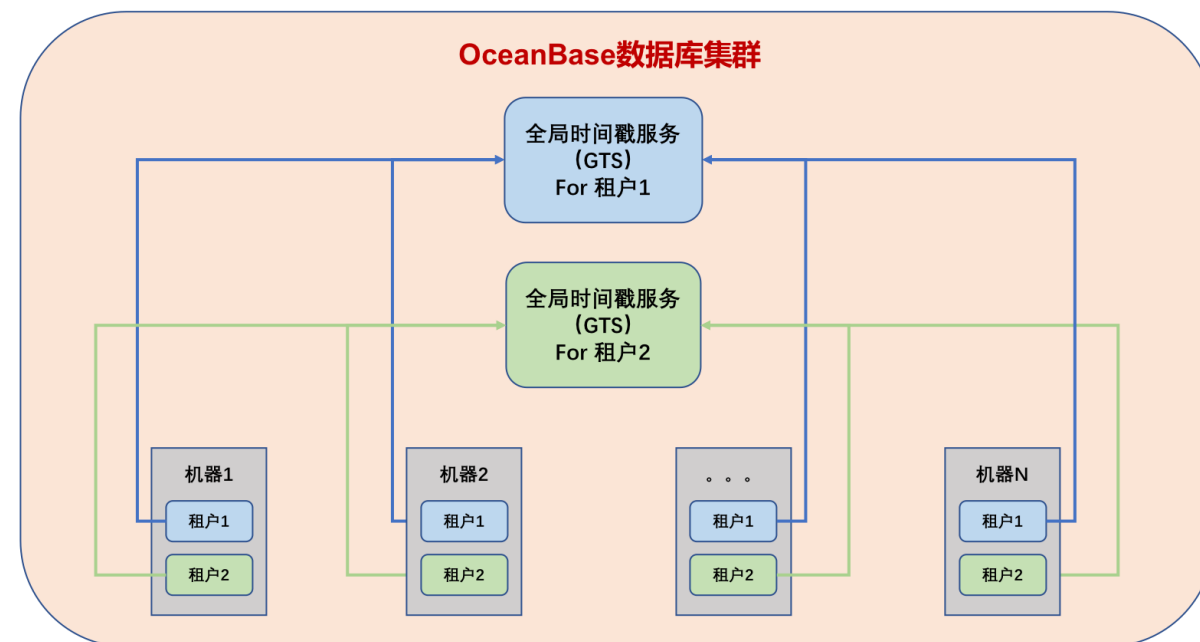
集中式服务：单系统时钟来源，多个事务共享。

按“**租户**”划分，管理灵活；将集群内事务分流到多个GTS，降低性能风险。

高性能：单节点每秒可响应**2百万次**服务请求。

低 Overhead：对性能整体影响**不超过5%**。

帮助OceanBase实现**全局一致性快照**，进一步**加强全局一致性功能**（如全局读写一致性，全局索引等）！

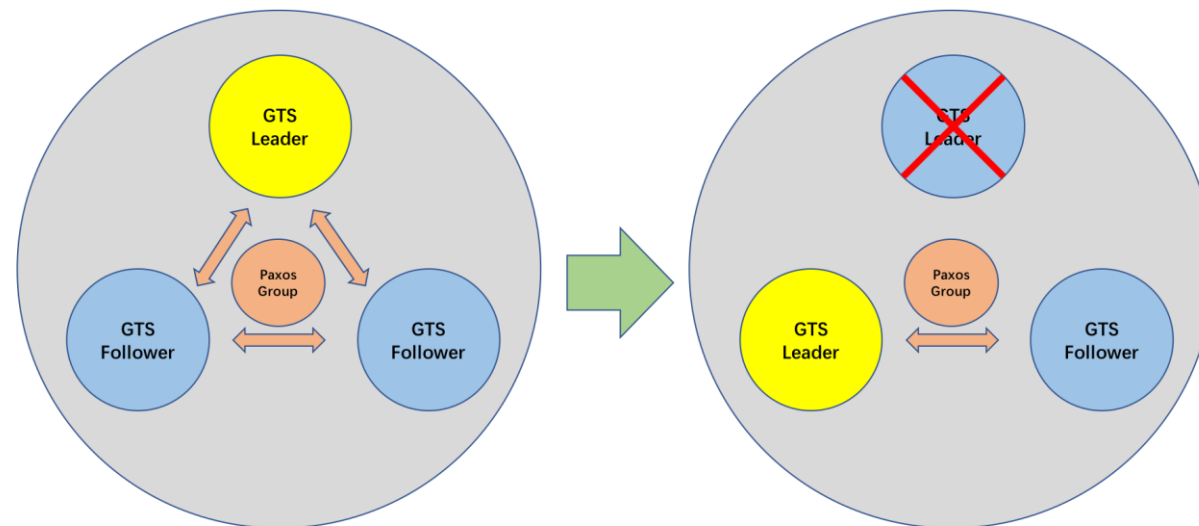


# GTS的可用性和可靠性

Paxos协议组：避免单点风险。

服务故障时自动回复（**1~15秒**），保障服务高可用。

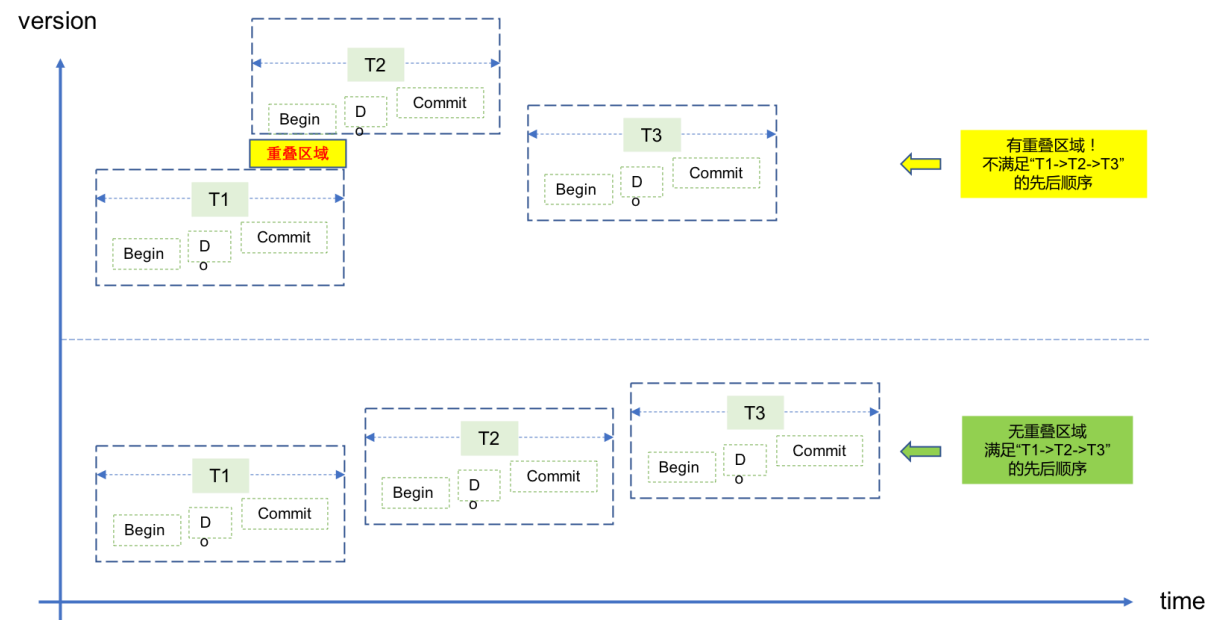
响应过慢时自动重发，避免事务处理被卡住。



# 网络抖动怎么办？

如果“**完整**”的事务之间有前后顺序，则GTS保证一致性。

如果事务之间有“**重叠**”，则事务本身不具备前后顺序，结果具有随机性。



# 谢谢

## THANK YOU

演讲人：邬腾



微信公众号  
OceanBase

关注OceanBase微信公众号  
回复关键词“1027” 获取PPT



OceanBase