

数据库系统 | Database Systems

第十课:并发控制

Lecture 10: concurrency control

Lecturer: Harbour

Date: 2022.01.26



Scan CC WeChat to Join the Community添加CC好友,接受进群邀请

Welcome to follow the GitHub repo

欢迎关注我们的代码仓库

https://github.com/cnosdb/cnosdb





计划教学内容



必讲

DB/DBMS

关系模型与关系代数

数据库存储

散列索引(哈希)

B+树

查询处理

并发控制

如果有兴趣

SQL

查询优化

恢复系统

分布式OLTP/OLAP

本节内容



- 原子性
- 一致性
- 隔离性
- 可串行化
- 持久性
- 两阶段锁

原子性



- Atomicity
- 事务中的所有动作都发生,或者不发生。

DBMS保证事务的执行结果只有两种:

- 在完成所有操作后 Commit。
- 在完成部分操作后主动或被动 Abort。

原子性



• 原子性的实现方式:

日志

影子分页

一致性



- Consistency
- **如果每个事**务是一致的,DB**开始**时也是一致的,那么它结束时也是一致的。
- 一致性是指系统从一个正确的状态(当前的状态满足预定的约束), 迁移到 另一个正确的状态。
- ACID就是说事务能够通过AID来保证这个C的过程。C是目的, AID都是手段。

隔离性



- Isolation
- 一个事务的执行与其他事务的执行是隔离的。
- DBMS需要限制数据的访问顺序(并发调度)。
- **隔离性的最高**标准是可串行化,即并发调度下数据库的结果与分别串行运行 这些事务所得的数据库完全一致。

可串行化



- 什么是冲突?
- 当两个操作同时满足以下三个条件时,我们认为它们是冲突的(conflicting):
 - 来自不同的事务
 - 对同一个对象操作
 - 两个操作至少有一个是 write 操作

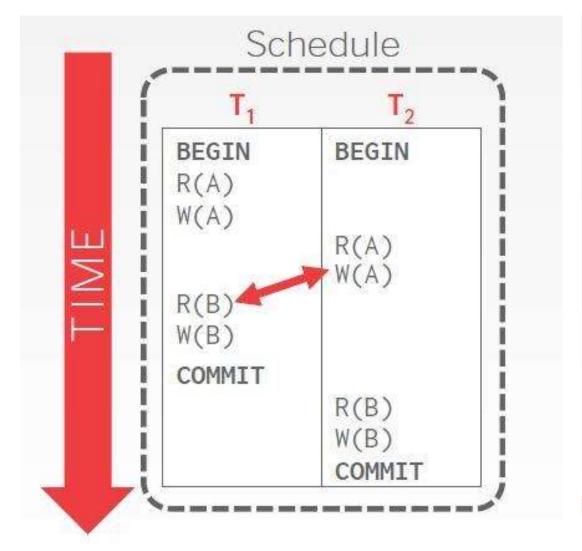
可串行化

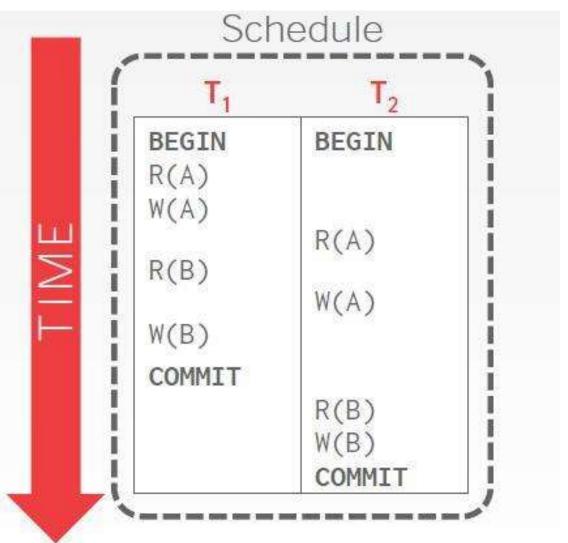


• 冲突的可串行化

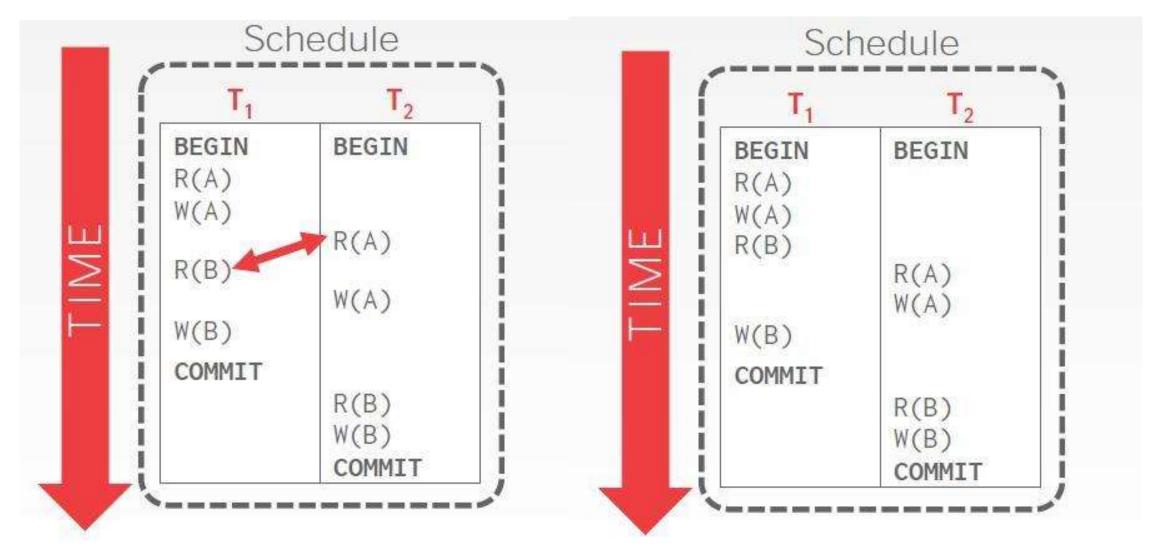
如果两个调度涉及相同事务的相同操作,并且其中一个调度经过一系列非冲突**指令的交**换后可以转换为另一个调度,则我们称它们为冲突等价(conflict equivalent)。**如果一个**调度**S与某个串行**调度是冲突等价的,则称 **S 是冲突可 串行化**(conflict serializable)。



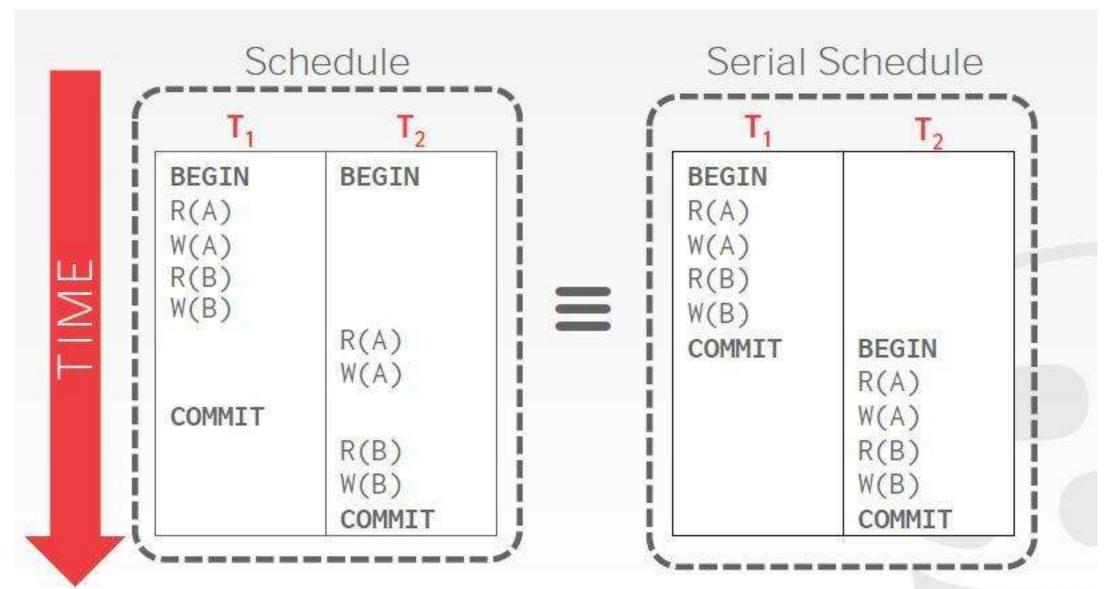




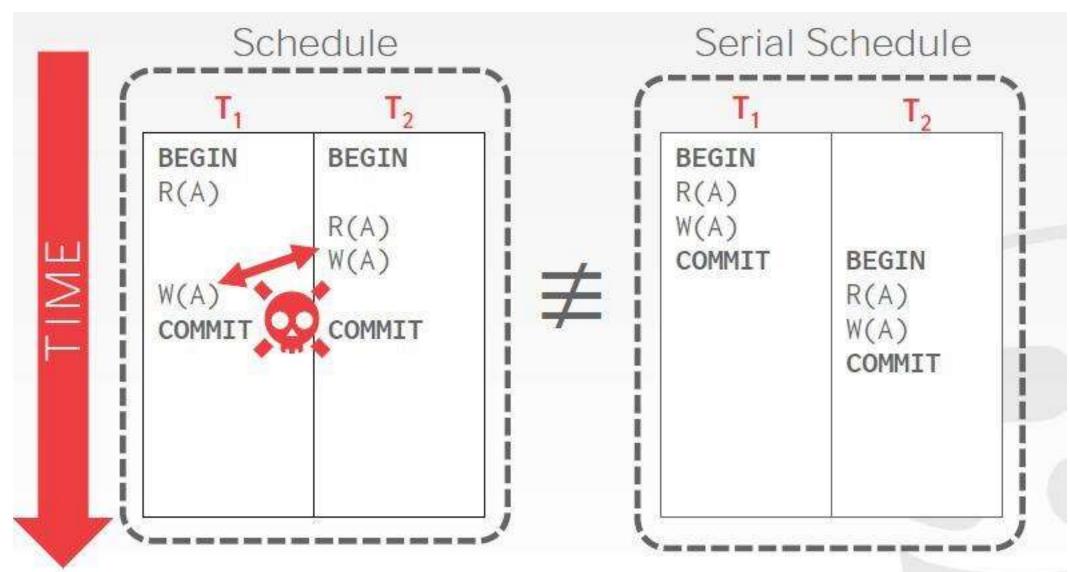












持久性



- **D**urability
- 如果一个事务被commit,则它的效果将保持不变。
- 一般情况下,事务提交的所有更改都应该是持久的。

72 两阶段锁



| | Locks | Latches |
|--------------------|--|-----------------------------|
| Separate | User transactions | Threads |
| Protect | Database Contents | In-Memory Data Structures |
| During | Entire Transactions | Critical Sections |
| Modes | Shared, Exclusive, Update, Intention | Read, Write |
| Handle deadlock by | Detection & Resolution Waits-for, Timeout, Aborts | Avoidance Coding Discipline |
| Kept in | Lock Manager | Protected Data Structure |



Q&A



Scan CC WeChat to Join the Community 添加CC好友,接受进群邀请

Welcome to follow the GitHub repo 欢迎关注我们的代码仓库 https://github.com/cnosdb/cnosdb



