

日志



胡卉芪
华东师范大学
数据科学与工程学院
hqhu@dase.ecnu.edu.cn

ACID与日志的关系

- A (Atomicity): 原子性
 - each transaction be "all or nothing"
- C (Consistency): 一致性
 - any transaction will bring the database from one valid state to another
- I (Isolation): 隔离性
 - the concurrent execution of transactions results in a system state that would be obtained if transactions were executed serially
- D (Durability): 持久性
 - once a transaction has been committed, it will remain so, even in the event of power loss, crashes, or errors

故障 (Failure)

- 存储介质故障 (Media failure)
- 灾难性故障 (Catastrophic failure)
- 系统故障 (System failure)
 - 断电
 - 软件中止
 - 操作系统中止
- *以上都可能使用日志恢复

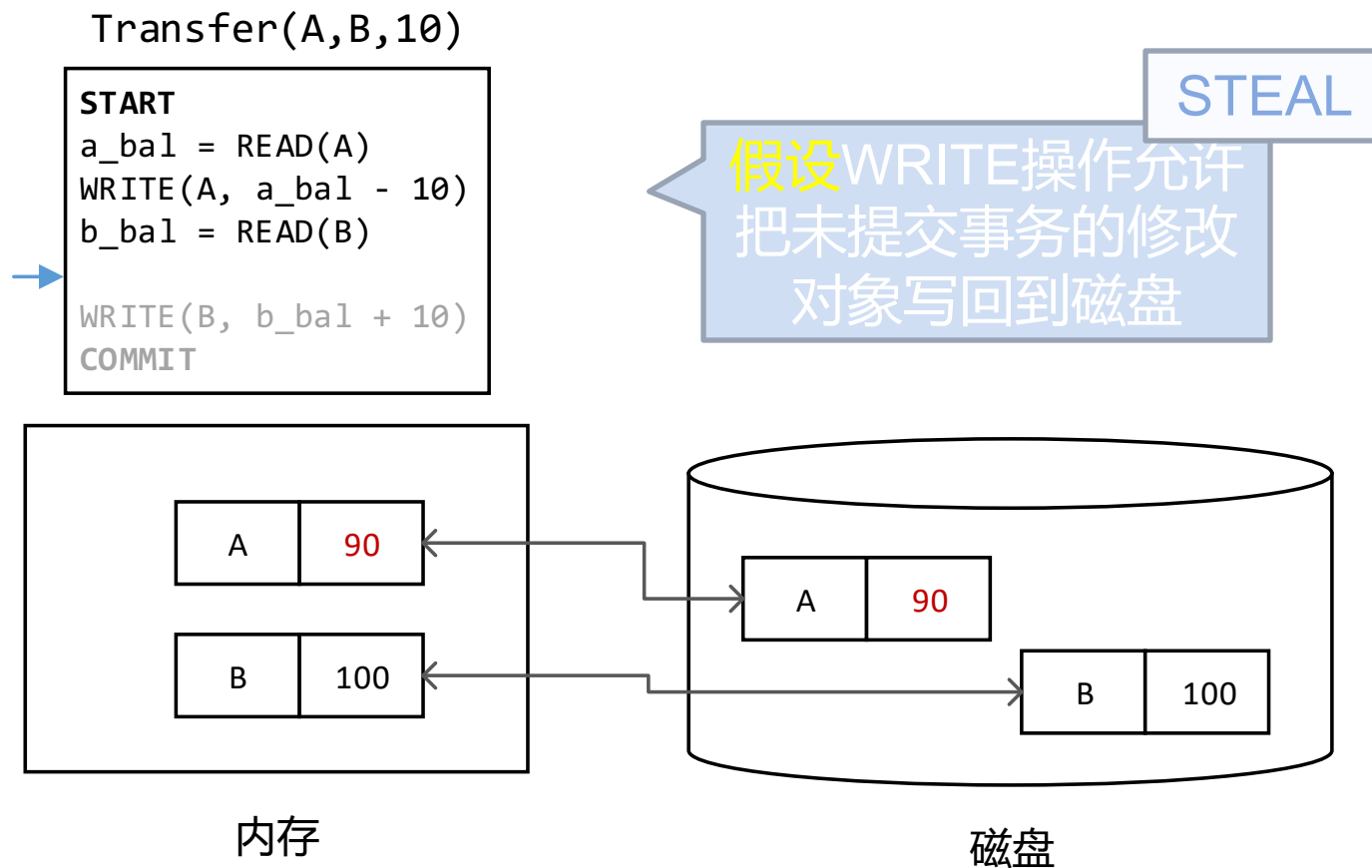
缓冲区策略与日志关系

缓冲区替换策略

- Steal
 - 允许未提交事务数据刷入磁盘
- Force
 - 只有事务的所有数据都刷入磁盘，事务才能提交

	No Steal	Steal
Force	Trivial	
No Force		Desired

Steal



Steal意味着Undo日志

Transfer(A,B,10)

START

a_bal = READ(A)

WRITE(A, a_bal - 10)

b_bal = READ(B)

WRITE(B, b_bal + 10)

COMMIT

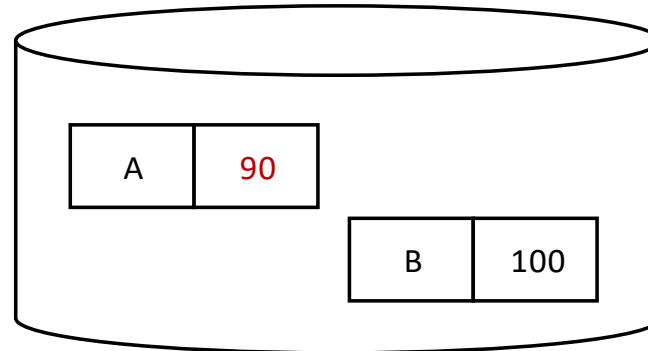
故障

用户 余额

A	90
B	100



内存



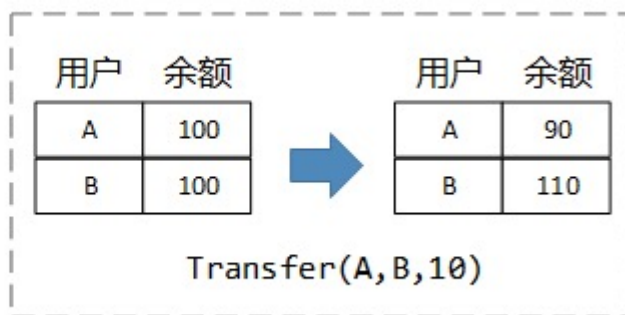
磁盘

Force

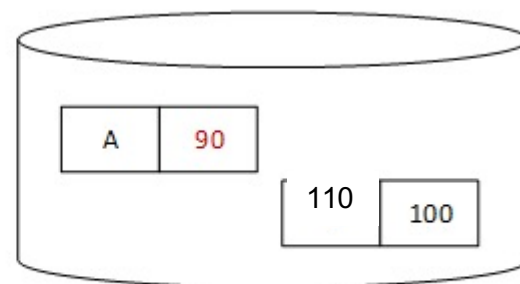
- 只有数据都落盘后事务才能提交
- No Force意味着已经提交的事务数据可能没有落盘
 - 需要Redo日志

Transfer(A,B,10)

```
START  
a_bal = READ(A)  
WRITE(A, a_bal - 10)  
b_bal = READ(B)  
WRITE(B, b_bal + 10)  
COMMIT
```



内存



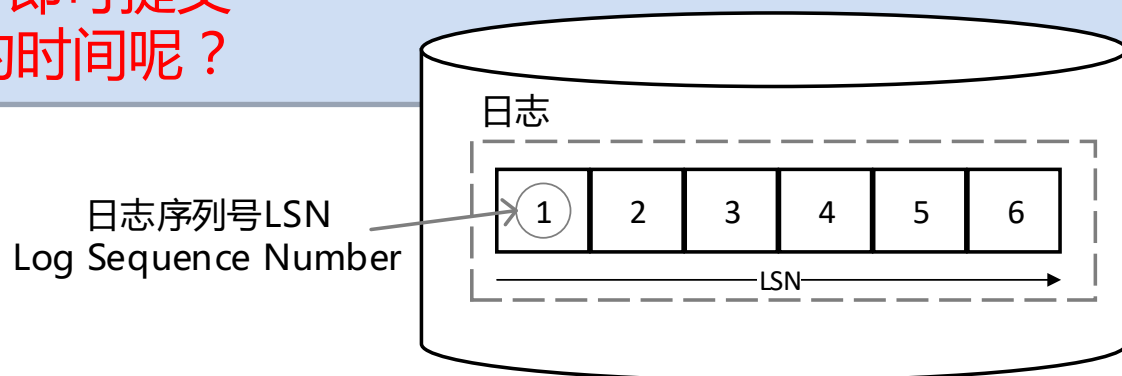
磁盘

WAL日志

概念

- 先写日志WAL (write-ahead logging)
 - 在写数据之前先写日志

1. 将对数据库的修改记录在单独的存储空间中（日志缓冲区）
日志缓冲区和数据缓冲区是两回事
2. 日志只支持追加操作（顺序I/O）
3. 修改的数据对象持久化之前，需要保证其对应的修改已记录在日志文件中（WAL）
4. 日志落盘后事务即可提交
S2PL释放锁的时间呢？

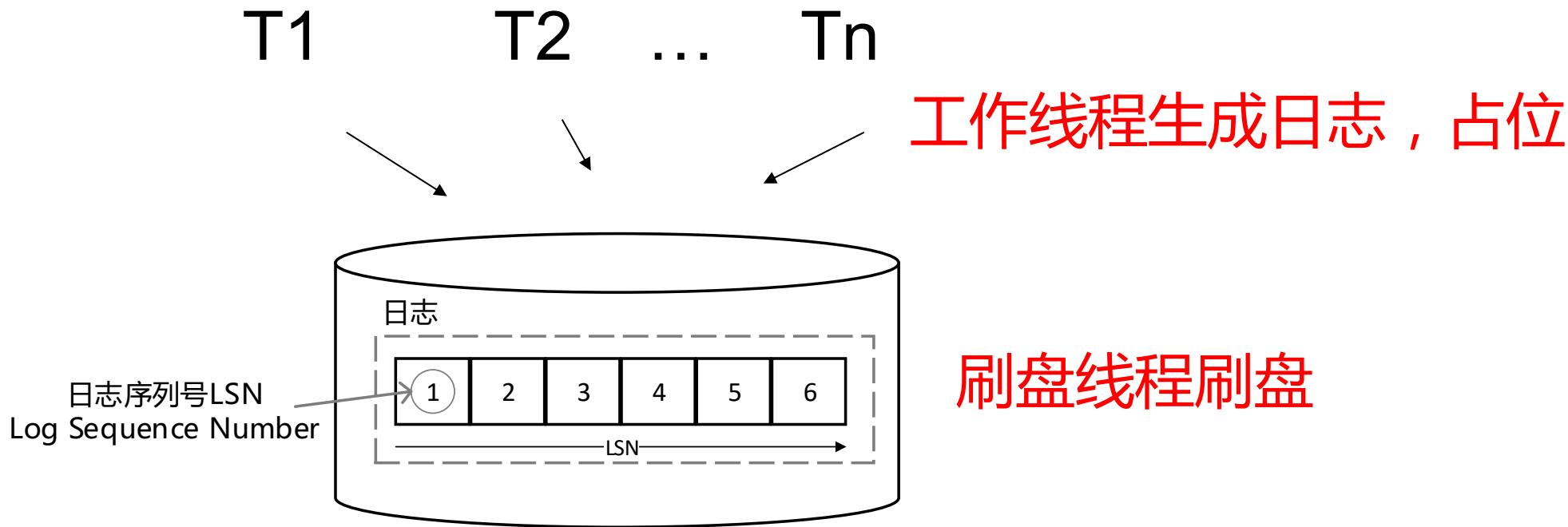


先写日志WAL

- 日志的主要步骤
 - 生成日志
 - 在缓冲区中占位（生成LSN）
 - 刷盘
 - 事务提交(可以理解为返回客户端，事务修改对外部可见)

先写日志WAL

WAL流程



刷盘阶段-组提交

- 每条日志刷盘，I/O代价较高
- 多个事务的日志一起按批次刷盘
- 组提交减少I/O次数
 - 日志量积累到一定大小
 - 定期

