

隔离级别与一致性级别的关系



并发控制与一致性协议在数据库系统中的作用

2

□ 并发控制

- 保证数据库充分利用多核资源并行执行事务的能力
 - 保证并发事务的多个操作之间不相互影响（事务的写入对并发事务的可见性）
 - 对并发事务间相互影响的约束
- 以隔离级别作为衡量标准

□ 一致性协议

- 维护多个数据副本，保证节点失败时持续服务的能力
 - 保持多个副本数据的一致，即要求多副本上事务执行顺序是一致的（事务的写入以何种顺序可见）
 - 对事务顺序的约束
- 以一致性级别作为衡量标准

并发控制和一致性协议共同作用于分布式数据库，才能保证数据库行为的正确性！

隔离级别

3

□ 常见的隔离级别

➤ **ANSI SQL标准(1992):** 基于异常定义

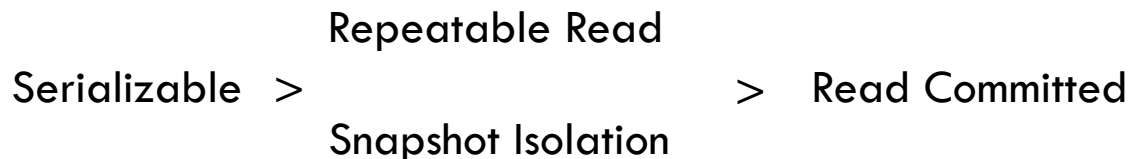
- 脏读、不可重复读、幻读
- 可串行化: 避免所有异常
- 可重复读: 避免脏读、不可重复读
- 读已提交: 避免脏读

➤ **A Critique of ANSI(1995):**

- 提出了丢失更新、写偏、读偏异常
- 定义了多版本下的隔离级别 快照隔离 (SI)

➤ **Atul Adya的博士论文 (Generalized Isolation Level Definitions) :**

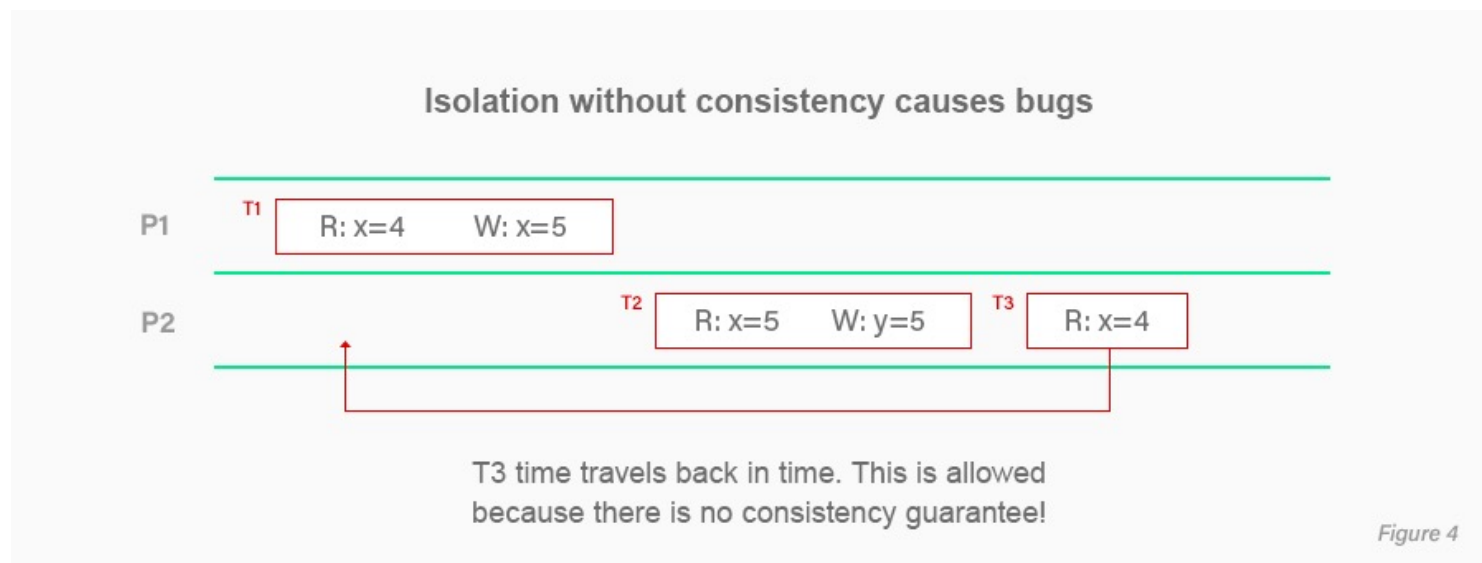
- 基于事务的依赖关系, 对隔离级别进行了精确定义



可串行化在分布式环境下的新异常

4

□ 一个仅保证可串行化的系统在多副本环境下是正确的吗？



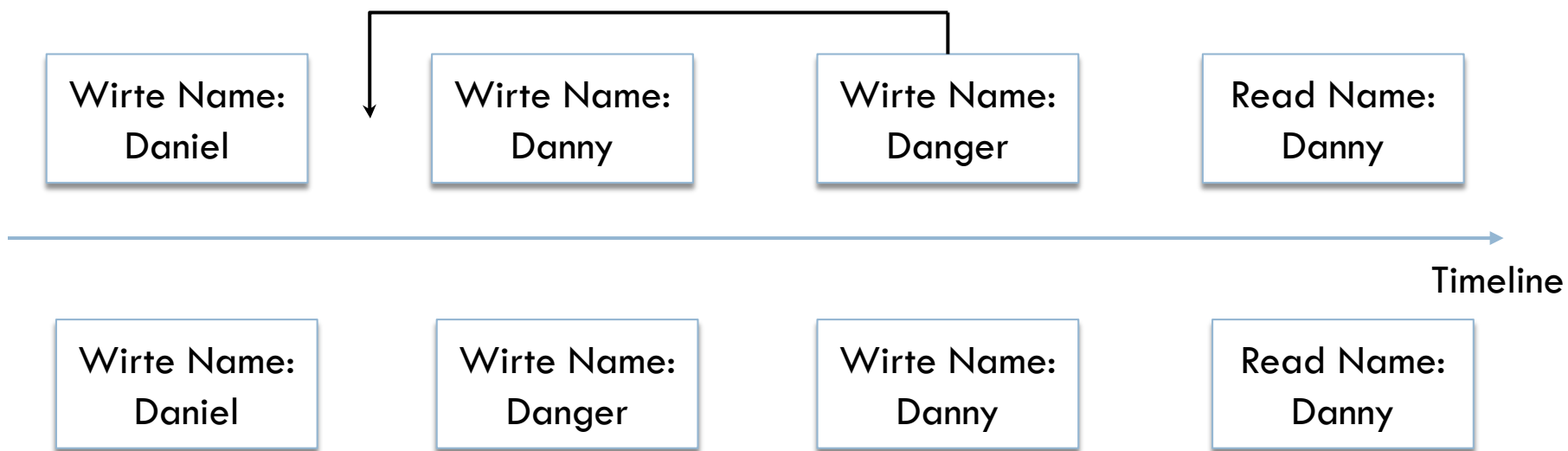
□ **Serializability**仅相当于**one copy serializability**，即每个数据项只有一份

□ 在分布式环境下出现异常的原因在于：

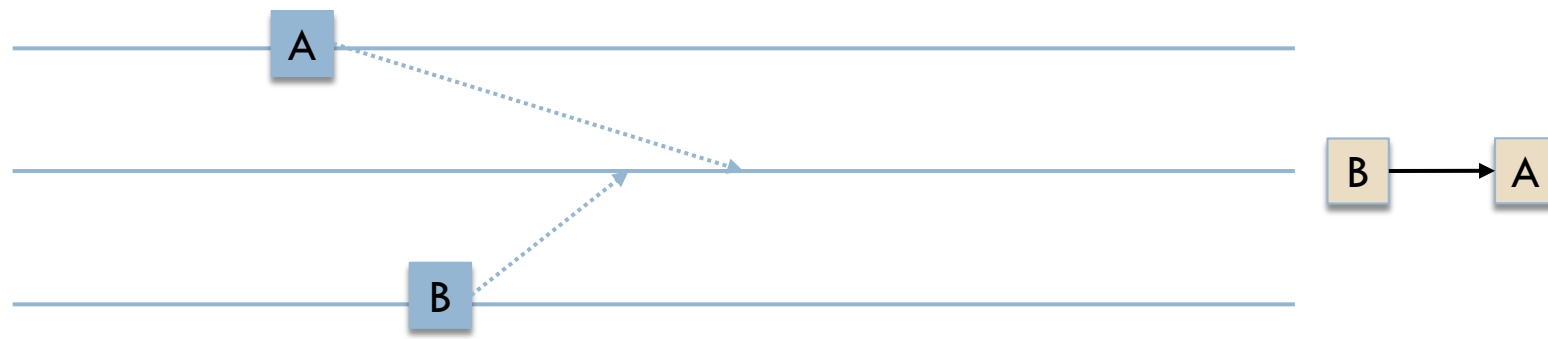
- 可串行化系统只保证一个串行的顺序，而对于具体的顺序并未加以限制
- 而分布式场景下，缺乏一个同步时钟，因此不同节点上操作顺序发生时间错乱，该现象称为“**time travel**”

Time Travel异常： 1. Immortal Write

5

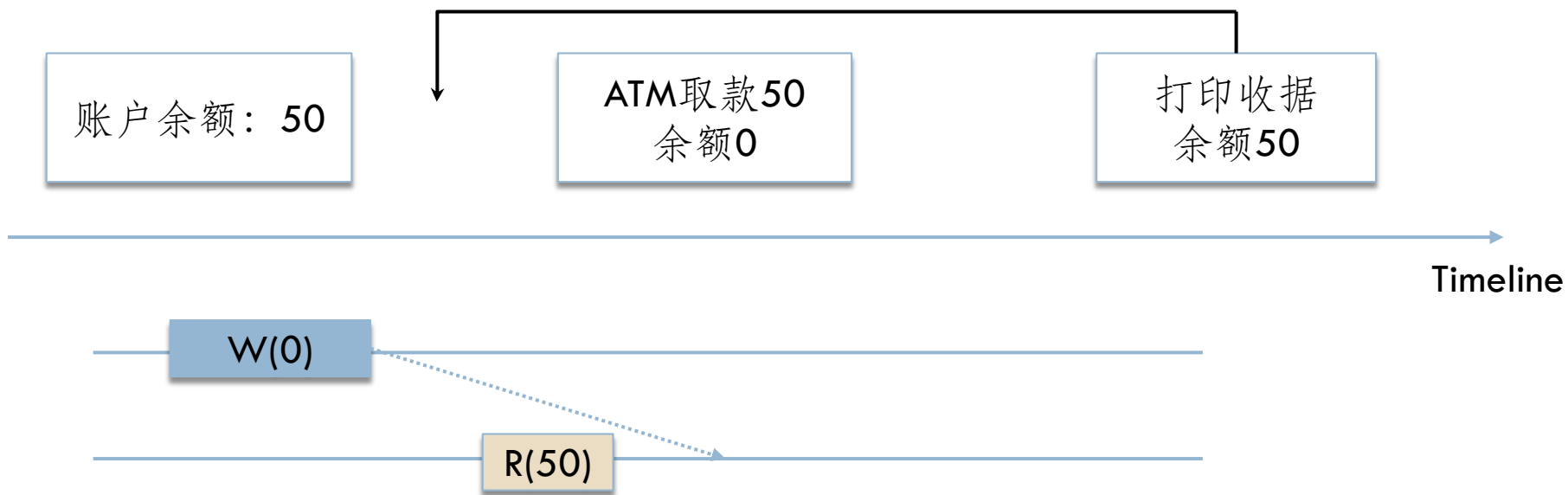


- ❑ **Immortal Write**本质上讲是一种写操作的时间错乱。
- ❑ 对于多主异步复制的数据库来说，该异常很容易发生。



Time Travel异常： 2. Stale Read

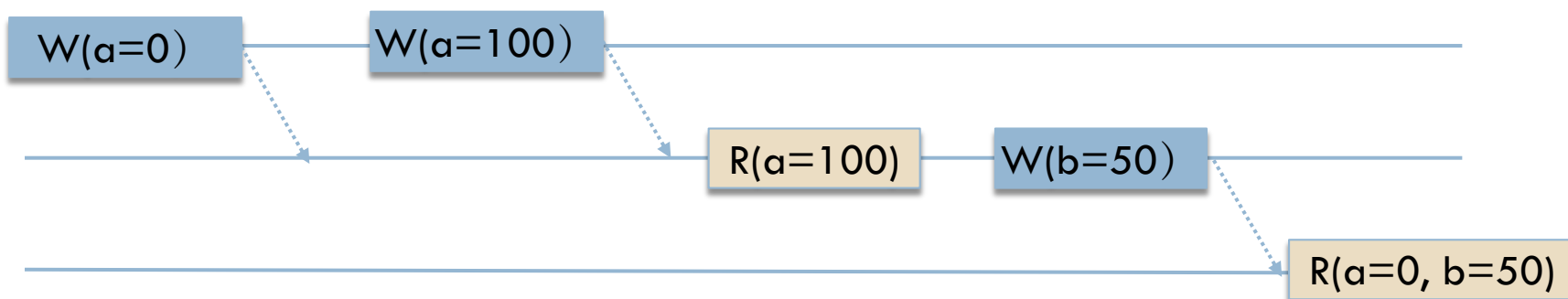
6



- ❑ **Steal Read**本质上讲是一种读写间的时间错乱。
- ❑ 对于异步复制的数据库来说，该异常很容易发生。
 - 读写不在同一副本，而读在写的结果复制完成前执行

Time Travel异常：3. Causal Reverse

7



□ 因果关系

- 同一进程内的操作
- **writes follow reads**
- 因果关系可传递

□ 具有因果关系的操作顺序发生反转

- 为了在分布式环境下保证数据库状态的正确性，除了保证可串行化以外，还要避免**Time Travel**异常，这就需要引入可线性化。
- 一致性级别对操作/事务的执行顺序提出了约束，可线性化是最理想的一致性级别。

一致性级别

8

- ❑ 一致性级别起源于共享内存中研究多核处理器不同线程写入顺序。类比到多副本系统中，也同样适用。
- ❑ **可串行化+可线性化 = 严格可串行化** (strict serializability)，是最高级别的正确性保证。
- ❑ 更强的一致性级别意味着更多的约束，是否能像隔离级别一样，对一致性加以放松？
- ❑ 操作间的顺序约束
 - **Real-time order**: 保持操作真实发生的物理时间顺序
 - **Total (same) order**: 操作是全局排序的，所有副本按照该顺序执行
 - **Causal order**: 保持具有因果关系的操作顺序，无关操作可乱序
 - **Process order**: 保持进程内操作的顺序

W(x=5)

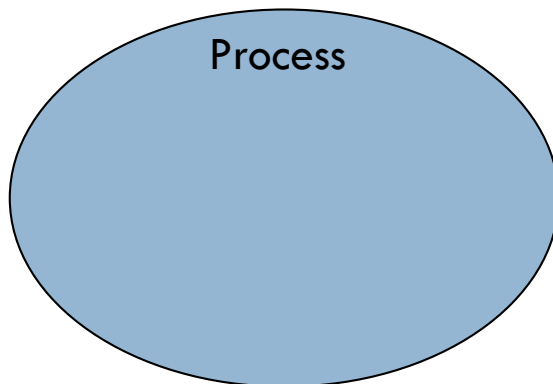
R(x=5)

W(y=10)

W(z=10)

几种Order的关系

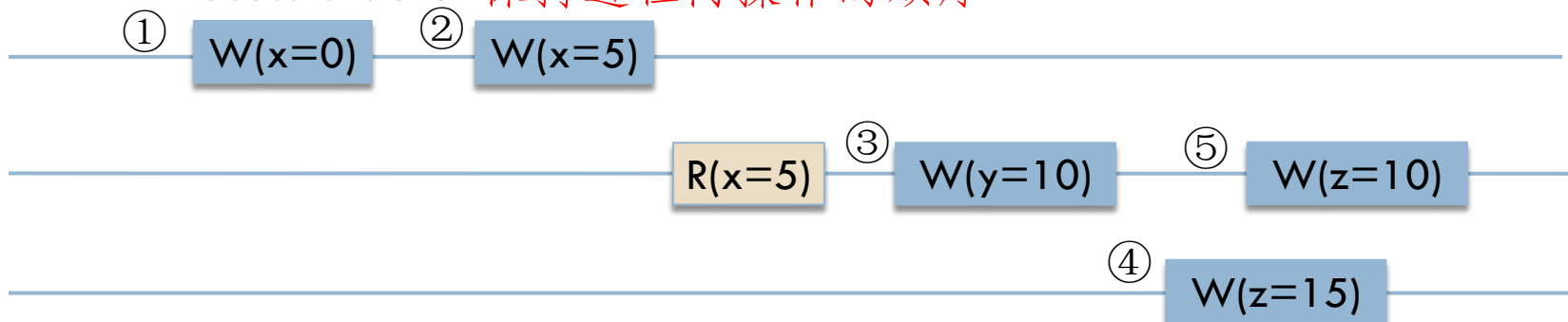
9



①②顺序必须保持，③⑤也一样

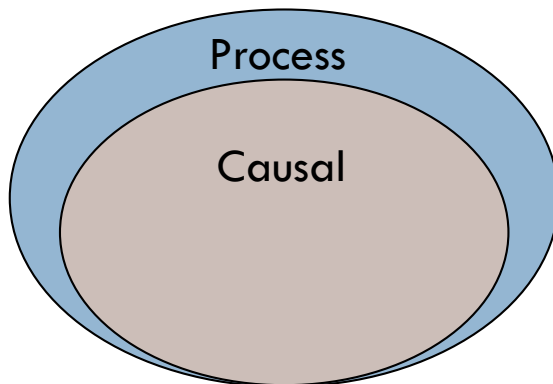
□ 操作间的顺序约束

- **Real-time order**: 保持操作真实发生的物理时间顺序
- **Total (same) order**: 操作是全局排序的，所有副本按照该顺序执行
- **Causal order**: 保持具有因果关系的操作顺序，无关操作可乱序
- **Process order**: 保持进程内操作的顺序



几种Order的关系

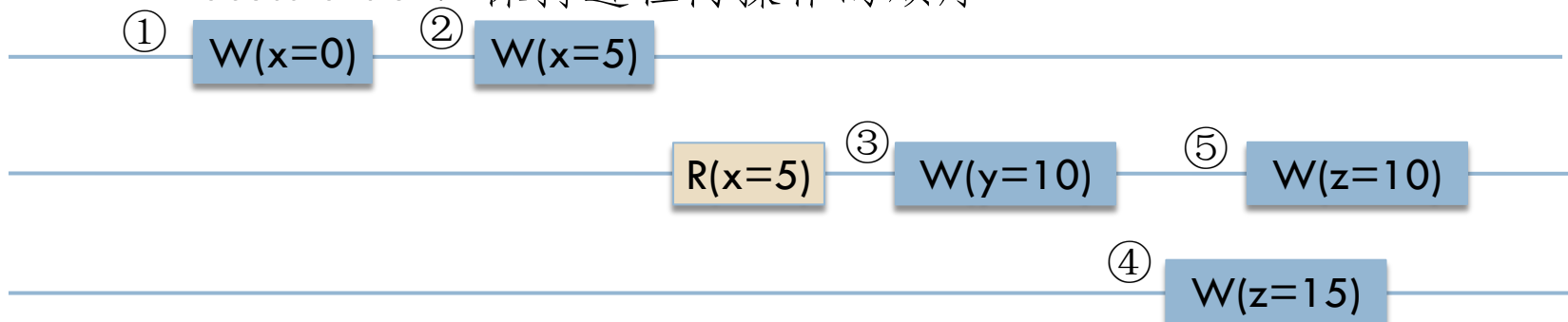
10



①②③⑤顺序必须保持

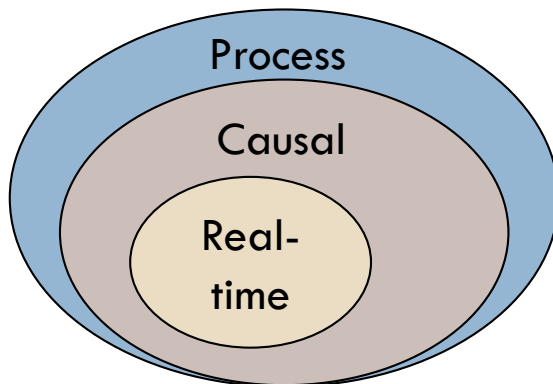
□ 操作间的顺序约束

- **Real-time order**: 保持操作真实发生的物理时间顺序
- **Total (same) order**: 操作是全局排序的，所有副本按照该顺序执行
- **Causal order**: 保持具有因果关系的操作顺序，无关操作可乱序
- **Process order**: 保持进程内操作的顺序



几种Order的关系

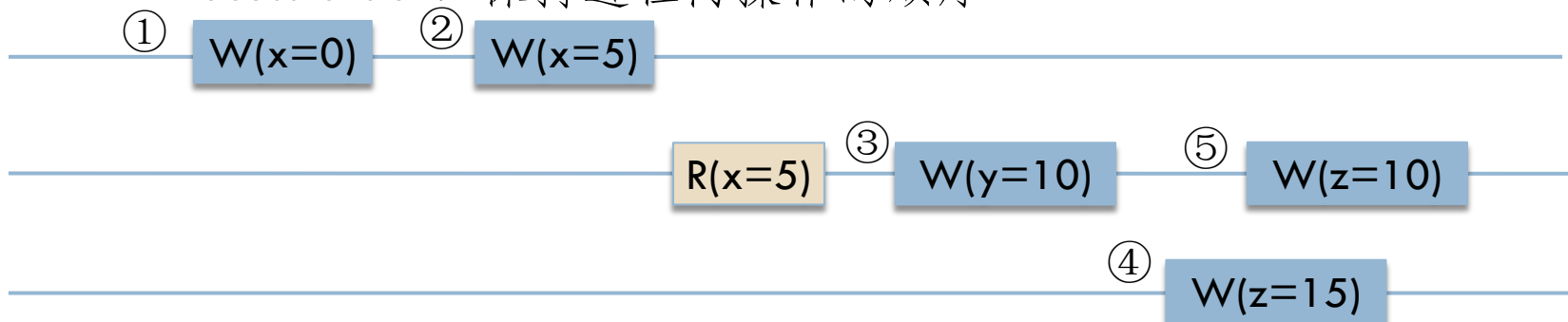
11



①②③顺序必须保持，
④⑤可乱序，但必须在③之后

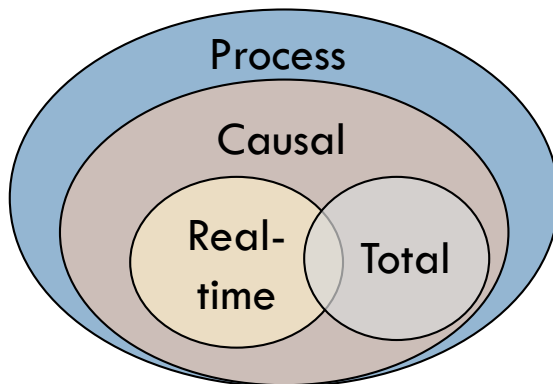
□ 操作间的顺序约束

- **Real-time order:** 保持操作真实发生的物理时间顺序
- **Total (same) order:** 操作是全局排序的，所有副本按照该顺序执行
- **Causal order:** 保持具有因果关系的操作顺序，无关操作可乱序
- **Process order:** 保持进程内操作的顺序



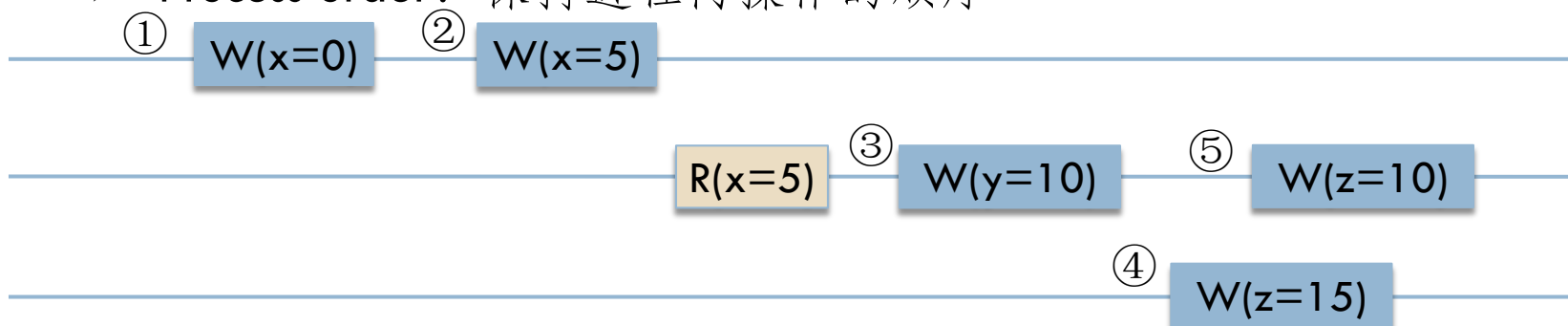
几种Order的关系

12



□ 操作间的顺序约束

- **Real-time order**: 保持操作真实发生的物理时间顺序
- **Total (same) order**: 操作是全局排序的，所有副本按照该顺序执行
- **Causal order**: 保持具有因果关系的操作顺序，无关操作可乱序
- **Process order**: 保持进程内操作的顺序



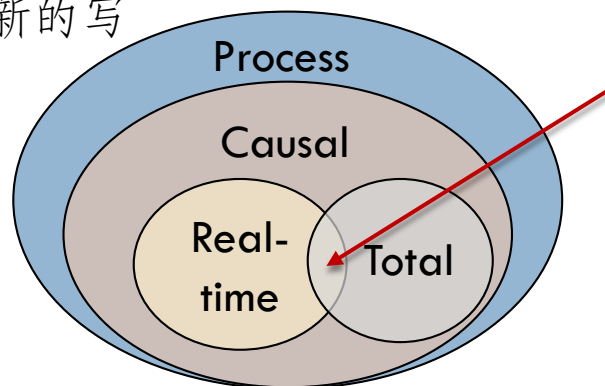
一致性级别——线性一致

13

□ 线性一致性 (linearizability)

- Real-time order ✓
- Total order ✓
- Causal order ✓
- Process order ✓

各节点上写生效顺序相同
写写先后顺序保持
读到最新的写



W(x=5)

W(y=10)

R(y=10)

R(x=5)

R(x=5)

R(y=10)

一致性级别——顺序一致

14

□ 顺序一致性 (sequential consistency) 各节点上写生效顺序相同

➤ Real-time order ✗

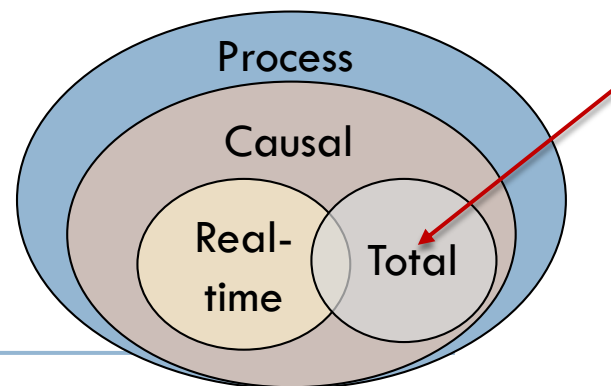
➤ Total order ✓

➤ Causal order ✓

➤ Process order ✓

写写顺序不必保持

不一定能读到最新的写



W(x=5)

W(y=10)

R(y=10)

R(x=0)

R(x=5)

R(y=10)

R(x=5)

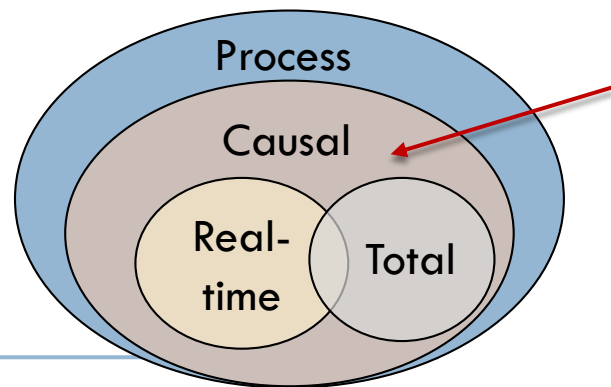
一致性级别——因果一致

15

□ 因果一致性 (causal consistency)

- Real-time order ✗
- Total order ✗
- Causal order ✓
- Process order ✓

因果相关的写生效顺序相同
无关写顺序不保持
不一定能读到最新的写



W(x=5)

R(x=5)

W(y=10)

R(y=10)

R(x=5)

R(x=5)

R(y=0)

隔离级别与一致性的结合

- 没有任何一致性保证的隔离保证并不是特别有用，分布式系统中需要两者结合共同保证数据库状态的正确性

