

数据库系统概论

An Introduction to Database System

第十章 数据库恢复技术

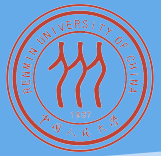
中国人民大学信息学院
陈红

第十章 数据库恢复技术



- 10.1 事务的基本概念
- 10.2 数据库恢复概述
- 10.3 故障的种类
- 10.4 恢复的实现技术
- 10.5 恢复策略**
- 10.6 具有检查点的恢复技术
- 10.7 数据库镜像
- 10.8 小结

10.5 恢复策略



10.5.1 事务故障的恢复

10.5.2 系统故障的恢复

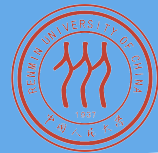
10.5.3 介质故障的恢复

10.5.1 事务故障的恢复



- ❖ **事务故障：事务在运行至正常终止点前被终止**
- ❖ **恢复方法**
 - 由恢复子系统利用日志文件撤消（ **UNDO** ）此事务已对数据库进行的修改
- ❖ **事务故障的恢复由系统自动完成，对用户是透明的，不需要用户干预**

事务故障的恢复步骤



1. 反向扫描文件日志（即从最后向前扫描日志文件），查找该事务的更新操作。
2. 对该事务的更新操作执行逆操作。即将日志记录中“更新前的值”写入数据库。
 - 插入操作，“更新前的值”为空，则相当于做删除操作
 - 删除操作，“更新后的值”为空，则相当于做插入操作
 - 若是修改操作，则相当于用修改前值代替修改后值

事务故障的恢复步骤



3. 继续反向扫描日志文件，查找该事务的其他更新操作，并做同样处理。
4. 如此处理下去，直至读到此事务的开始标记，事务故障恢复就完成了。

10.5 恢复策略



10.5.1 事务故障的恢复

10.5.2 系统故障的恢复

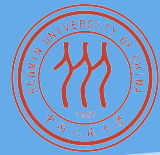
10.5.3 介质故障的恢复

10.5.2 系统故障的恢复



- ❖ 系统故障造成数据库不一致状态的原因
 - 未完成事务对数据库的更新已写入数据库
 - 已提交事务对数据库的更新还留在缓冲区没来得及写入数据库
- ❖ 恢复方法
 - 1. Undo 故障发生时未完成的事务
 - 2. Redo 已完成的事务
- ❖ 系统故障的恢复由系统在重新启动时自动完成，不需要用户干预

系统故障的恢复步骤



1. 正向扫描日志文件（即从头扫描日志文件）

- 重做 (REDO) 队列：在故障发生前已经提交的事务
 - 这些事务既有 **BEGIN TRANSACTION** 记录，也有 **COMMIT** 记录
- 撤销 (Undo) 队列：故障发生时尚未完成的事务
 - 这些事务只有 **BEGIN TRANSACTION** 记录，无相应的 **COMMIT** 记录

系统故障的恢复步骤



2. 对撤销 (Undo) 队列事务进行撤销 (UNDO) 处理
 - 反向扫描日志文件，对每个 UNDO 事务的更新操作执行逆操作
 - 即将日志记录中“更新前的值”写入数据库
3. 对重做 (Redo) 队列事务进行重做 (REDO) 处理
 - 正向扫描日志文件，对每个 REDO 事务重新执行登记的操作
 - 即将日志记录中“更新后的值”写入数据库

10.5 恢复策略



10.5.1 事务故障的恢复

10.5.2 系统故障的恢复

10.5.3 介质故障的恢复

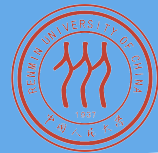
10.5.3 介质故障的恢复



1. 重装数据库

2. 重做已完成的事务

介质故障的恢复（续）

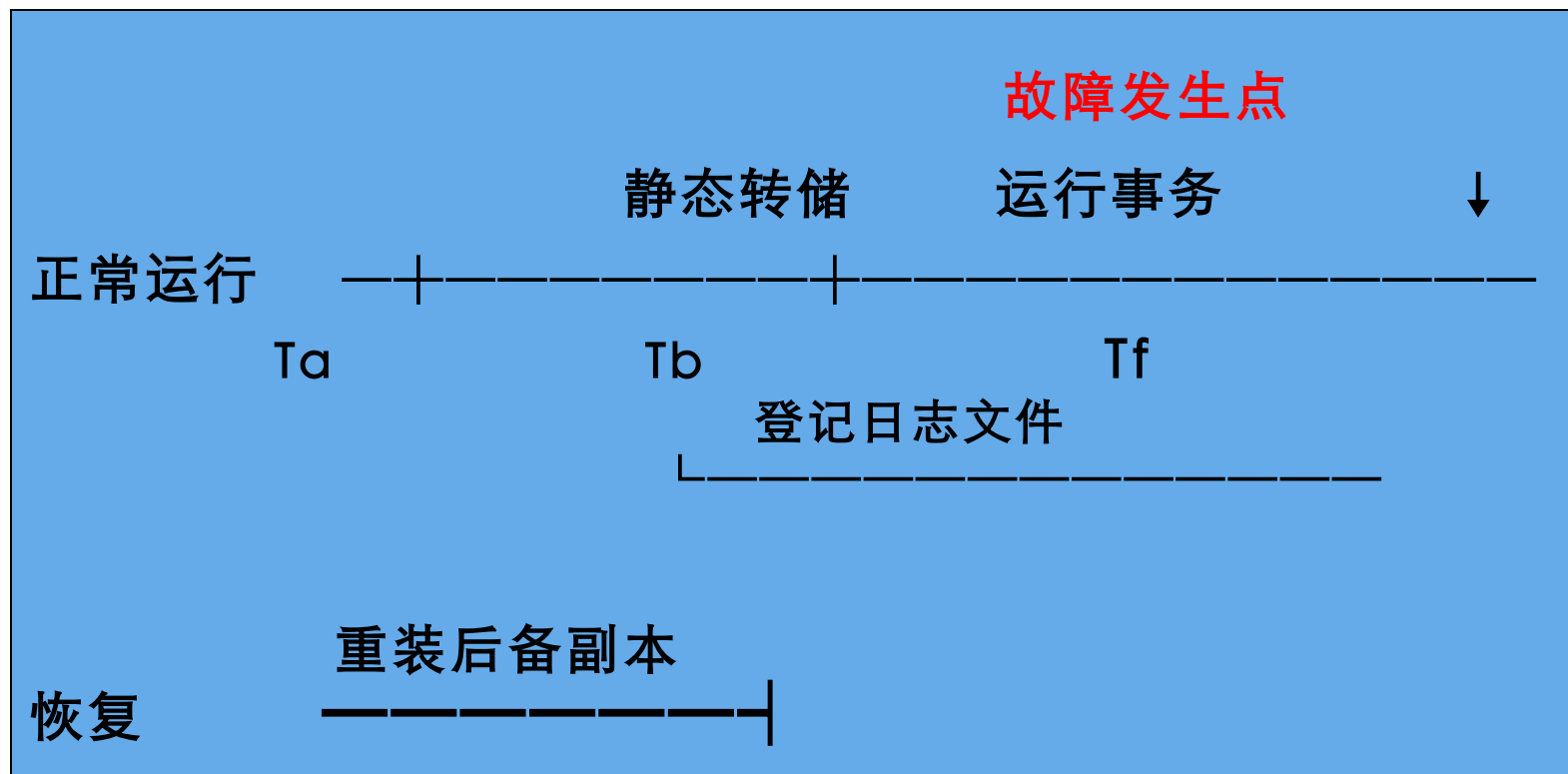


❖ 恢复步骤

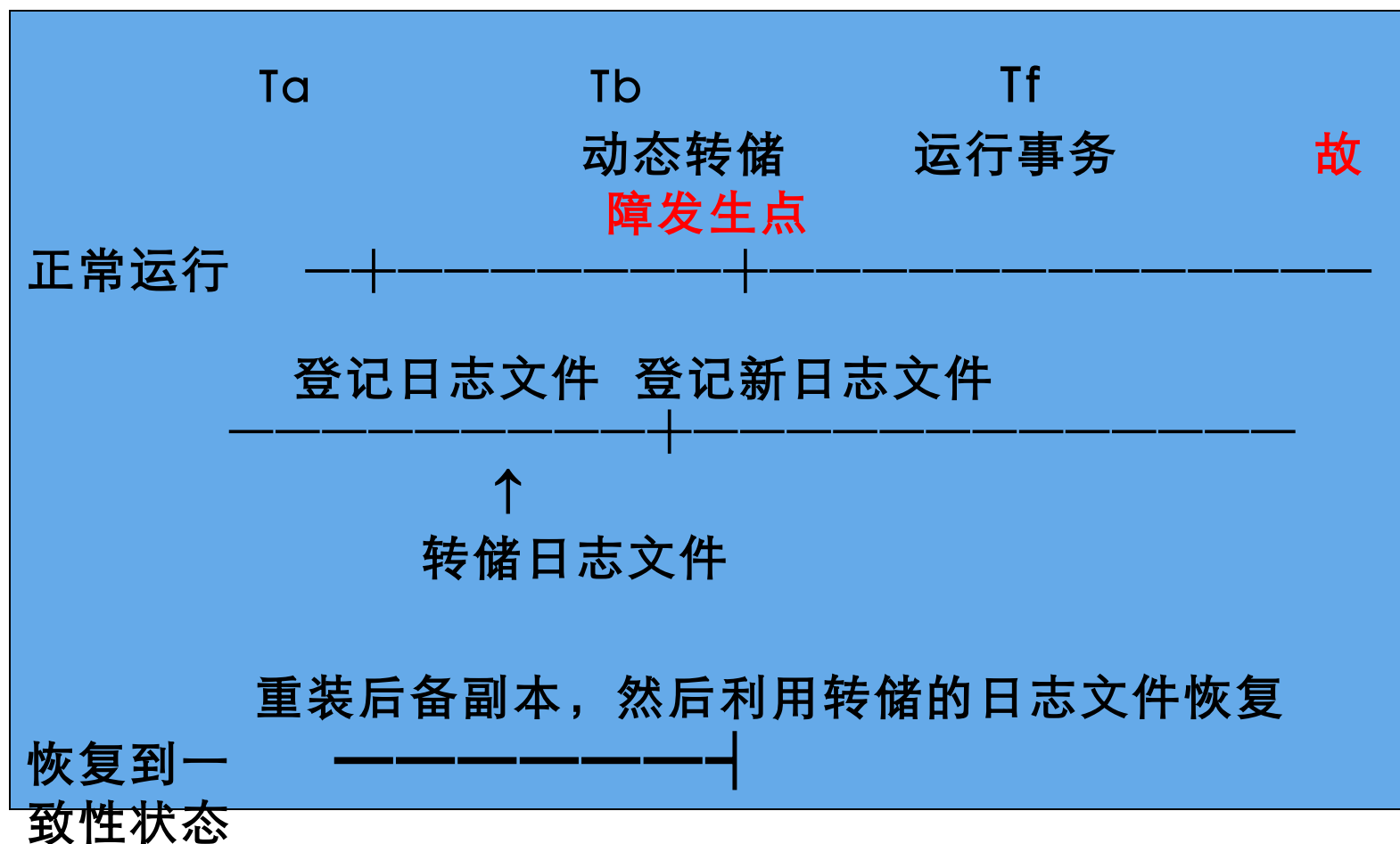
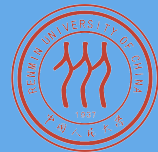
1. 装入最新的后备数据库副本（离故障发生时刻最近的转储副本），使数据库恢复到最近一次转储时的一致性状态。

- 对于静态转储的数据库副本，装入后数据库即处于一致性状态
- 对于动态转储的数据库副本，还须同时装入转储时刻的日志文件副本，利用与恢复系统故障的方法（即 REDO+UNDO），才能将数据库恢复到一致性状态。

利用静态转储副本将数据库恢复到一致性状态



利用动态转储副本将数据库恢复到一致性状态



介质故障的恢复（续）



2. 装入有关的日志文件副本（转储结束时刻的日志文件副本），重做已完成的事务。

- 首先扫描日志文件，找出故障发生时已提交的事务的标识，将其记入重做队列。
- 然后正向扫描日志文件，对重做队列中的所有事务进行重做处理。即将日志记录中“更新后的值”写入数据库。

介质故障的恢复（续）



介质故障的恢复需要 DBA 介入

❖ DBA 的工作

- 重装最近转储的数据库副本和有关的各日志文件副本
- 执行系统提供的恢复命令

❖ 具体的恢复操作仍由 DBMS 完成

第十章 数据库恢复技术



- 10.1 事务的基本概念
- 10.2 数据库恢复概述
- 10.3 故障的种类
- 10.4 恢复的实现技术
- 10.5 恢复策略
- 10.6 具有检查点的恢复技术
- 10.7 数据库镜像
- 10.8 小结

10.6 具有检查点的恢复技术



一、问题的提出

二、检查点技术

三、利用检查点的恢复策略

一、问题的提出



❖ 两个问题

- 搜索整个日志将耗费大量的时间
- REDO 处理：重新执行，浪费了大量时间

解决方案



❖ 具有检查点（checkpoint）的恢复技术

- 在日志文件中增加检查点记录（checkpoint）
- 增加重新开始文件
- 恢复子系统在登录日志文件期间动态地维护日志

二、检查点技术



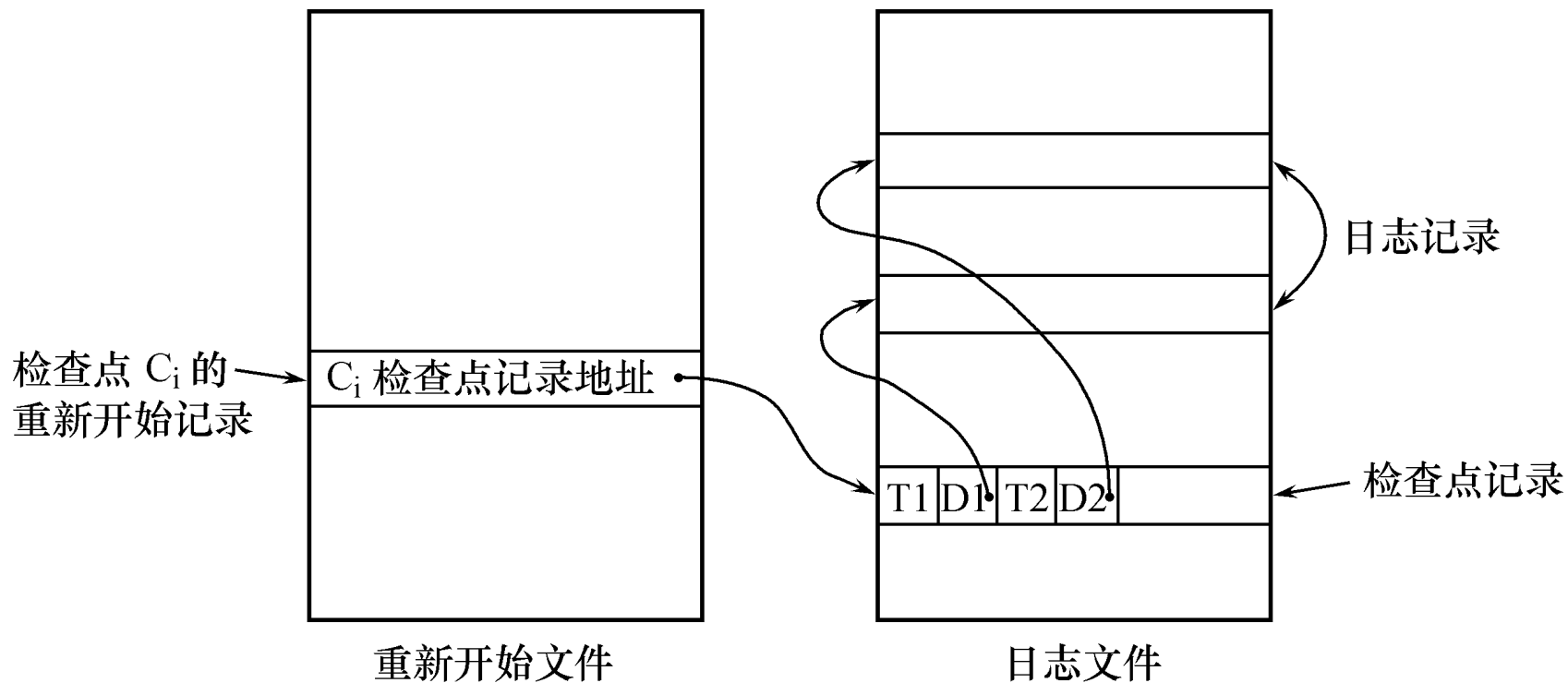
❖ 检查点记录的内容

- 1. 建立检查点时刻所有正在执行的事务清单
- 2. 这些事务最近一个日志记录的地址

❖ 重新开始文件的内容

- 记录各个检查点记录在日志文件中的地址

检查点技术（续）



具有检查点的日志文件和重新开始文件

动态维护日志文件的方法



❖ 动态维护日志文件的方法

周期性地执行如下操作：建立检查点，保存数据库状态。

具体步骤是：

- 1. 将当前 **日志** 缓冲区中的所有日志记录写入磁盘的日志文件上
- 2. 在日志文件中写入一个检查点记录
- 3. 将当前 **数据** 缓冲区的所有数据记录写入磁盘的数据库中
- 4. 把检查点记录在日志文件中的地址写入一个重新开始文

建立检查点



❖ 恢复子系统可以定期或不定期地建立检查点，保存数据库状态

■ 定期

➤ 按照预定的一个时间间隔，如每隔一小时建立一个检查点

■ 不定期

➤ 按照某种规则，如日志文件已写满一半建立一个检查点

三、利用检查点的恢复策略



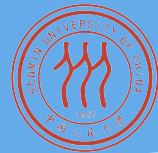
❖ 使用检查点方法可以改善恢复效率

- 当事务 T 在一个检查点之前提交

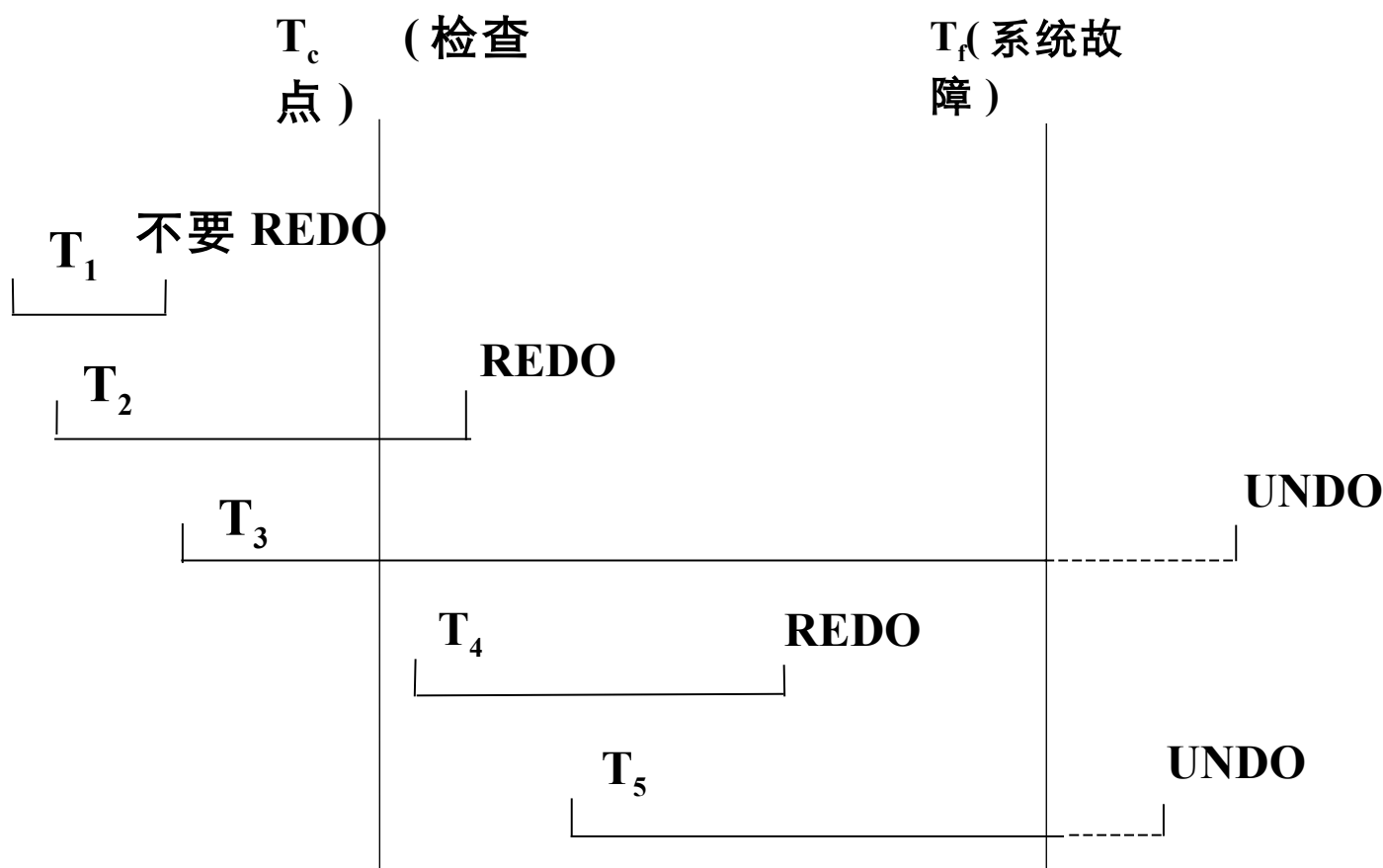
T 对数据库所做的修改已写入数据库

- 写入时间是在这个检查点建立之前或在这个检查点建立之时
- 在进行恢复处理时，没有必要对事务 T 执行 REDO 操作

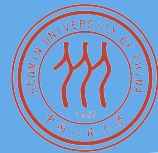
利用检查点的恢复策略（续）



系统出现故障时，恢复子系统将根据事务的不同状态采取不同的恢复策略



利用检查点的恢复策略（续）



- T1：在检查点之前提交
- T2：在检查点之前开始执行，在检查点之后故障点之前提交
- T3：在检查点之前开始执行，在故障点时还未完成
- T4：在检查点之后开始执行，在故障点之前提交
- T5：在检查点之后开始执行，在故障点时还未完成

恢复策略：

- T3 和 T5 在故障发生时还未完成，所以予以撤销
- T2 和 T4 在检查点之后才提交，它们对数据库所做的修改在故障发生时可能还在缓冲区中，尚未写入数据库，所以要 REDO

- T1 在检查点之前已提交，所以不必执行 REDO 操作

利用检查点的恢复步骤



1. 从重新开始文件中找到最后一个检查点记录在日志文件中的地址，由该地址在日志文件中找到最后一个检查点记录

利用检查点的恢复策略（续）



2. 由该检查点记录得到检查点建立时刻所有正在执行的事务清单 **ACTIVE-LIST**
 - 建立两个事务队列
 - **UNDO-LIST**
 - **REDO-LIST**
 - 把 **ACTIVE-LIST** 暂时放入 **UNDO-LIST** 队列， **REDO** 队列暂为空。

利用检查点的恢复策略（续）



3. 从检查点开始正向扫描日志文件，直到日志文件结束

- 如有新开始的事务 T_i ，把 T_i 暂时放入 UNDO-LIST 队列
- 如有提交的事务 T_j ，把 T_j 从 UNDO-LIST 队列移到 REDO-LIST 队列

4. 对 UNDO-LIST 中的每个事务执行 UNDO 操作

对 REDO-LIST 中的每个事务执行 REDO 操作

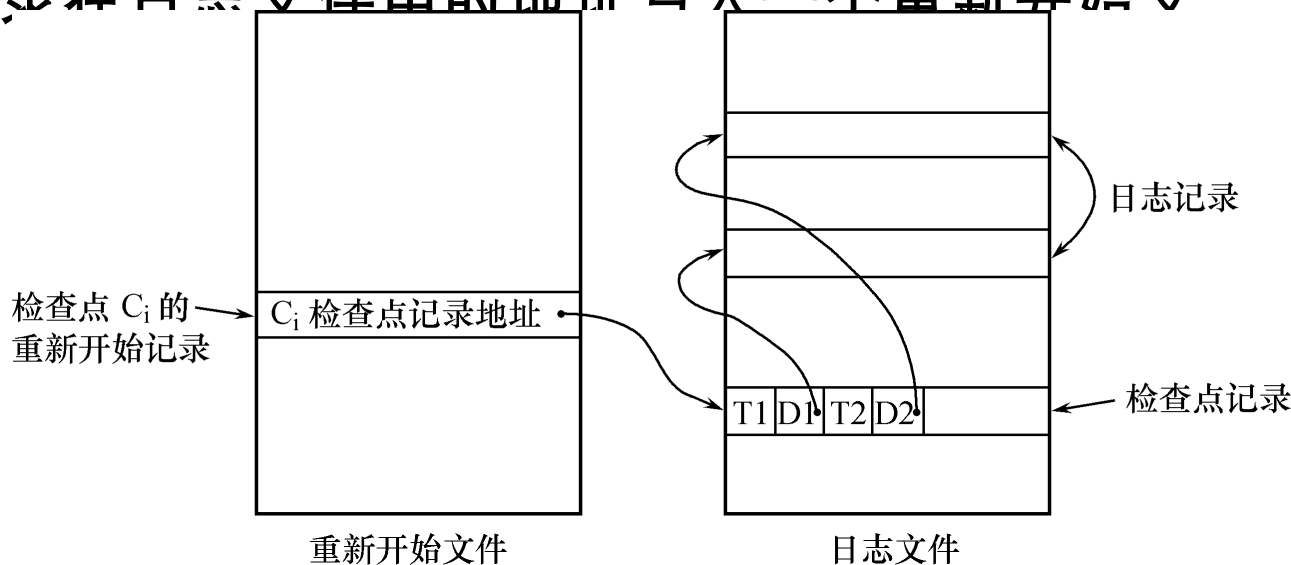
思考



❖ 检查点过程

- 1. 将当前日志缓冲区中的所有日志记录写入磁盘的日志文件上
- 2. 在日志文件中写入一个检查点记录
- 3. 将当前数据缓冲区的所有数据记录写入磁盘的数据库中
- 4. 把检查点记录在日志文件中的地址写入一个重新开始文件

如果在第2步之后
发生故障，会有问题吗？

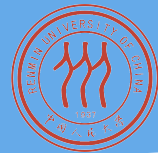


第十章 数据库恢复技术



- 10.1 事务的基本概念
- 10.2 数据库恢复概述
- 10.3 故障的种类
- 10.4 恢复的实现技术
- 10.5 恢复策略
- 10.6 具有检查点的恢复技术
- 10.7 数据库镜像
- 10.8 小结

10.7 数据库镜像



- ❖ 介质故障是对系统影响最为严重的一种故障，严重影响数据库的可用性
 - 介质故障恢复比较费时
 - 为预防介质故障，**DBA** 必须周期性地转储数据库

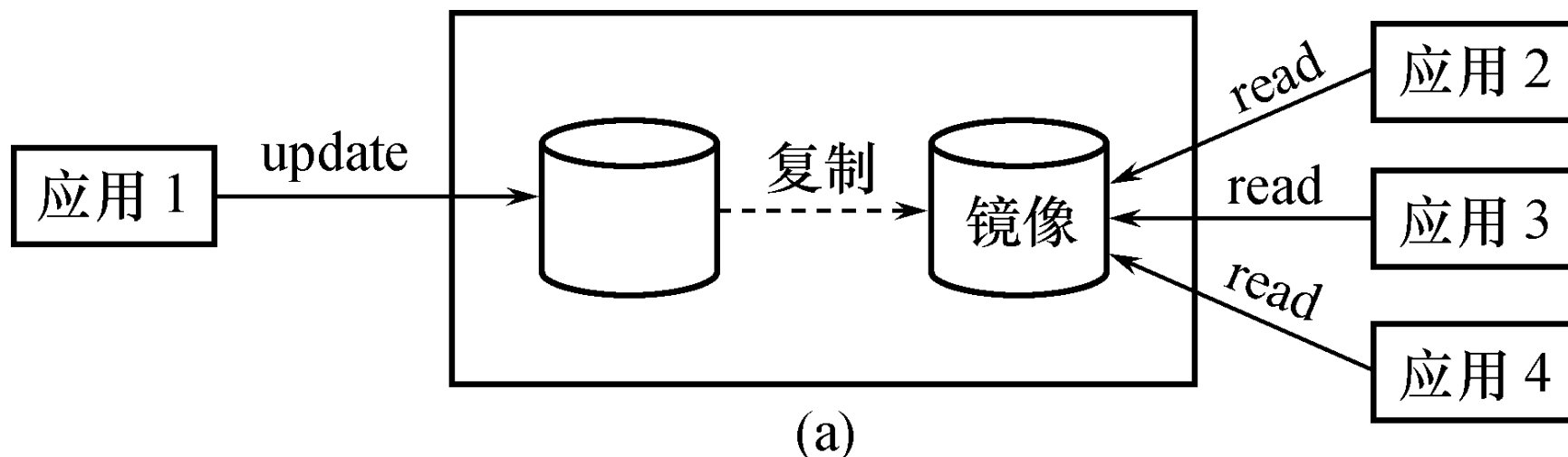
- ❖ 提高数据库可用性的解决方案
 - 数据库镜像（**Mirror**）

数据库镜像（续）



❖ 数据库镜像

- **DBMS** 自动把整个数据库或其中的关键数据复制到另一个磁盘上
- **DBMS** 自动保证镜像数据与主数据的一致性
每当主数据库更新时，**DBMS** 自动把更新后的数据复制过去

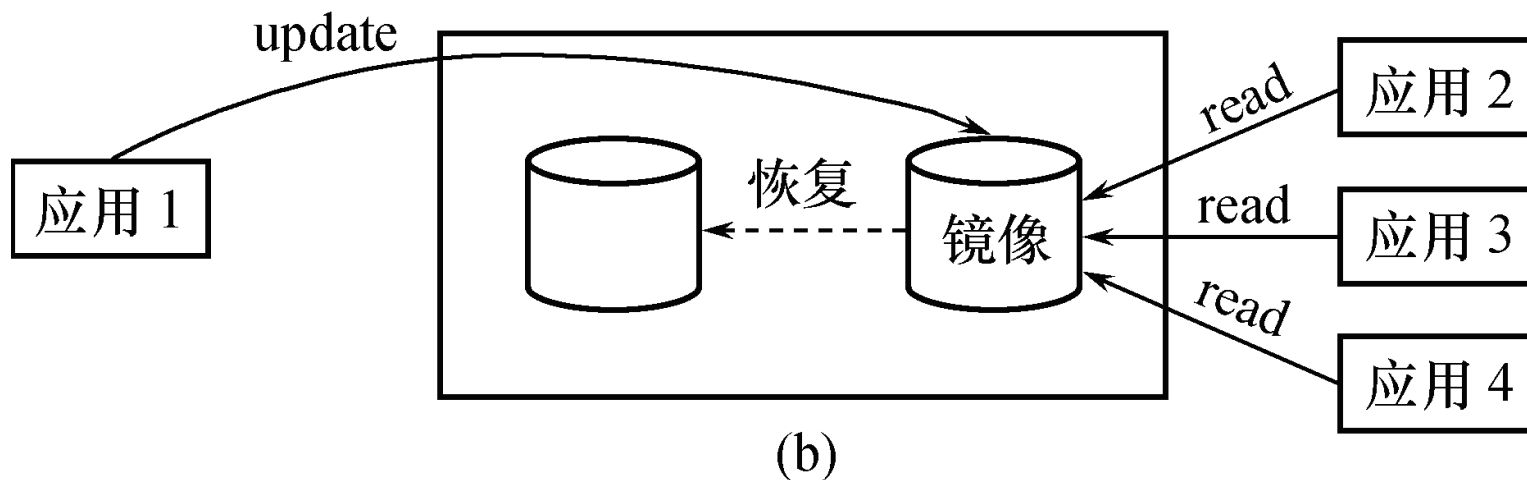


数据库镜像的用途

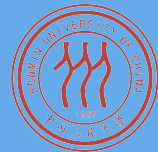


❖ 出现介质故障时

- 可由镜像磁盘继续提供使用
- 同时 **DBMS** 自动利用镜像磁盘数据进行数据库的恢复



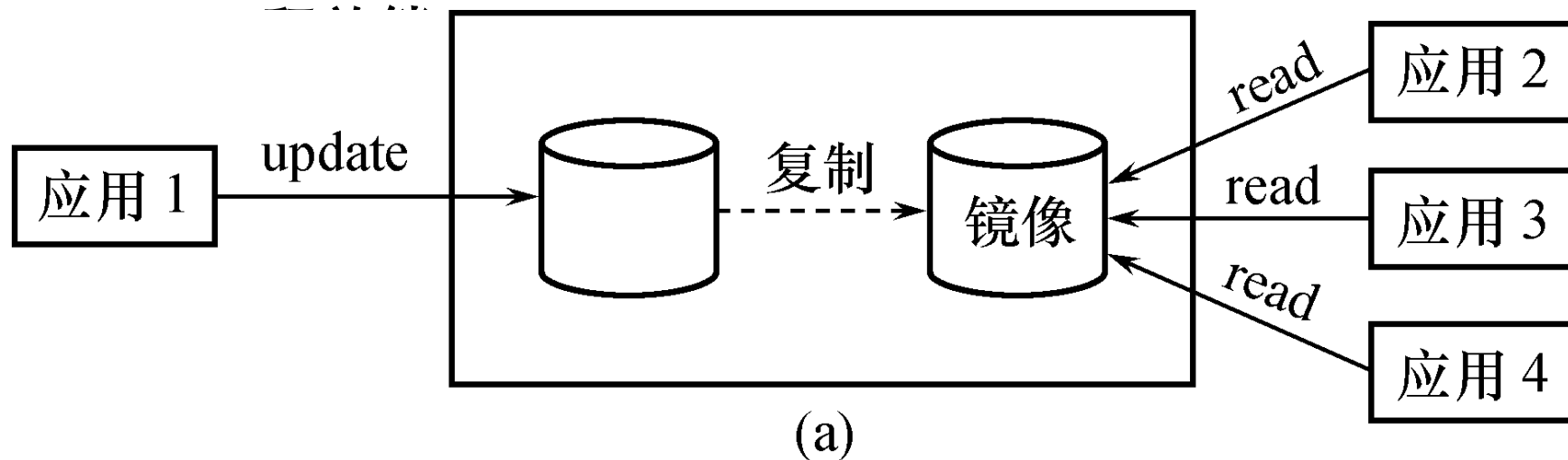
数据库镜像（续）



❖ 没有出现故障时

- 可用于并发操作

- 一个用户对数据加排他锁修改数据，其他用户可以读镜像数据库上的数据，而不必等待该用户



数据库镜像（续）



❖ 频繁地复制数据自然会降低系统运行效率

- 在实际应用中用户往往只选择对**关键数据**和**日志文件**镜像
- 不是对整个数据库进行镜像

第十章 数据库恢复技术



- 10.1 事务的基本概念
- 10.2 数据库恢复概述
- 10.3 故障的种类
- 10.4 恢复的实现技术
- 10.5 恢复策略
- 10.6 具有检查点的恢复技术
- 10.7 数据库镜像
- 10.8 小结

10.8 小结



❖ 事务的概念和性质

- 事务是数据库的逻辑工作单位
- DBMS 保证系统中一切事务的原子性、一致性、隔离性和持续性

小结（续）



❖ 故障的种类

- 事务故障
- 系统故障
- 介质故障

❖ 恢复中最经常使用的技术

- 数据库转储
- 登记日志文件

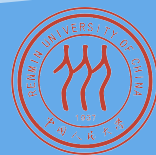
小结（续）



❖ 恢复策略

- 事务故障的恢复
 - UNDO
- 系统故障的恢复
 - UNDO + REDO
- 介质故障的恢复
 - 重装备份并恢复到一致性状态 + REDO

小结（续）



❖ 提高恢复效率的技术

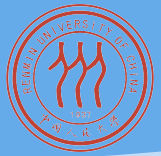
■ 检查点技术

- 可以提高系统故障的恢复效率
- 可以在一定程度上提高利用动态转储备份进行介质故障恢复的效率

■ 镜像技术

- 镜像技术可以改善介质故障的恢复效率

作业



❖ 笔头作业

P289-290 2,6,7,11,14

❖ 思考作业

P289-290 其他题，特别是 P290 9,10



