



2025年春季学期

数据库系统概论

An Introduction to Database Systems

第十一章 数据库恢复技术

中国科学技术大学

人工智能与数据科学学院

黄振亚, huangzhy@ustc.edu.cn



第十一章 数据库恢复技术

63

11.1 事务的基本概念

11.2 数据库恢复概述

11.3 故障的种类

11.4 恢复的实现技术

11.5 恢复策略

11.6 具有检查点的恢复技术

11.7 数据库镜像

11.8 小结



11.6 具有检查点的恢复技术

64

- 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

65

- 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
- 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（快速找到检查点）
 - 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

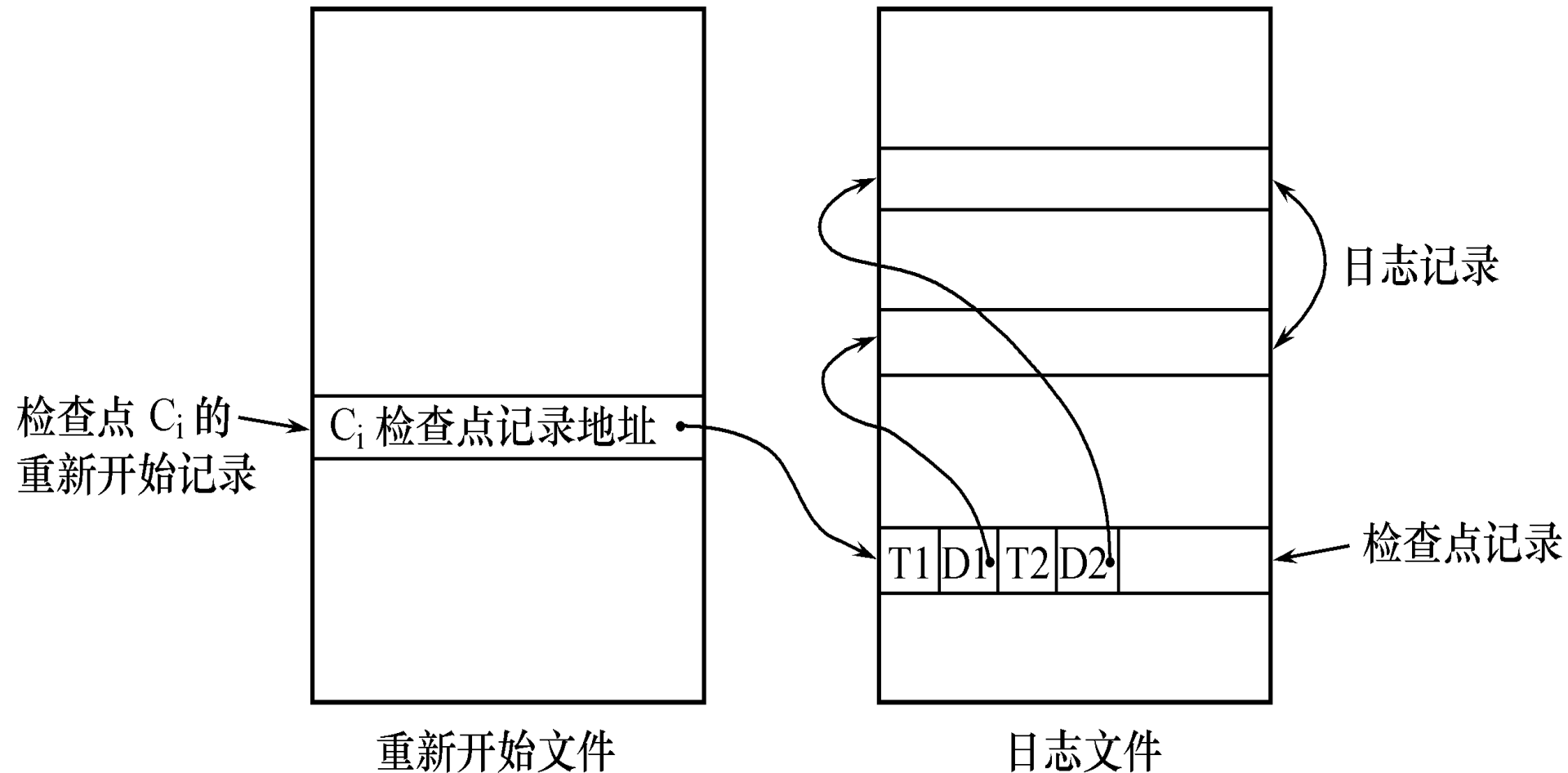
66

- 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）

67



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

68

□ 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.写日志缓冲区：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.记录检查点：在日志文件中写入一个检查点记录
- 3.写数据：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.记重新开始文件：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



建立检查点

69

- 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（通常是DBMS自动做）
 - 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

70

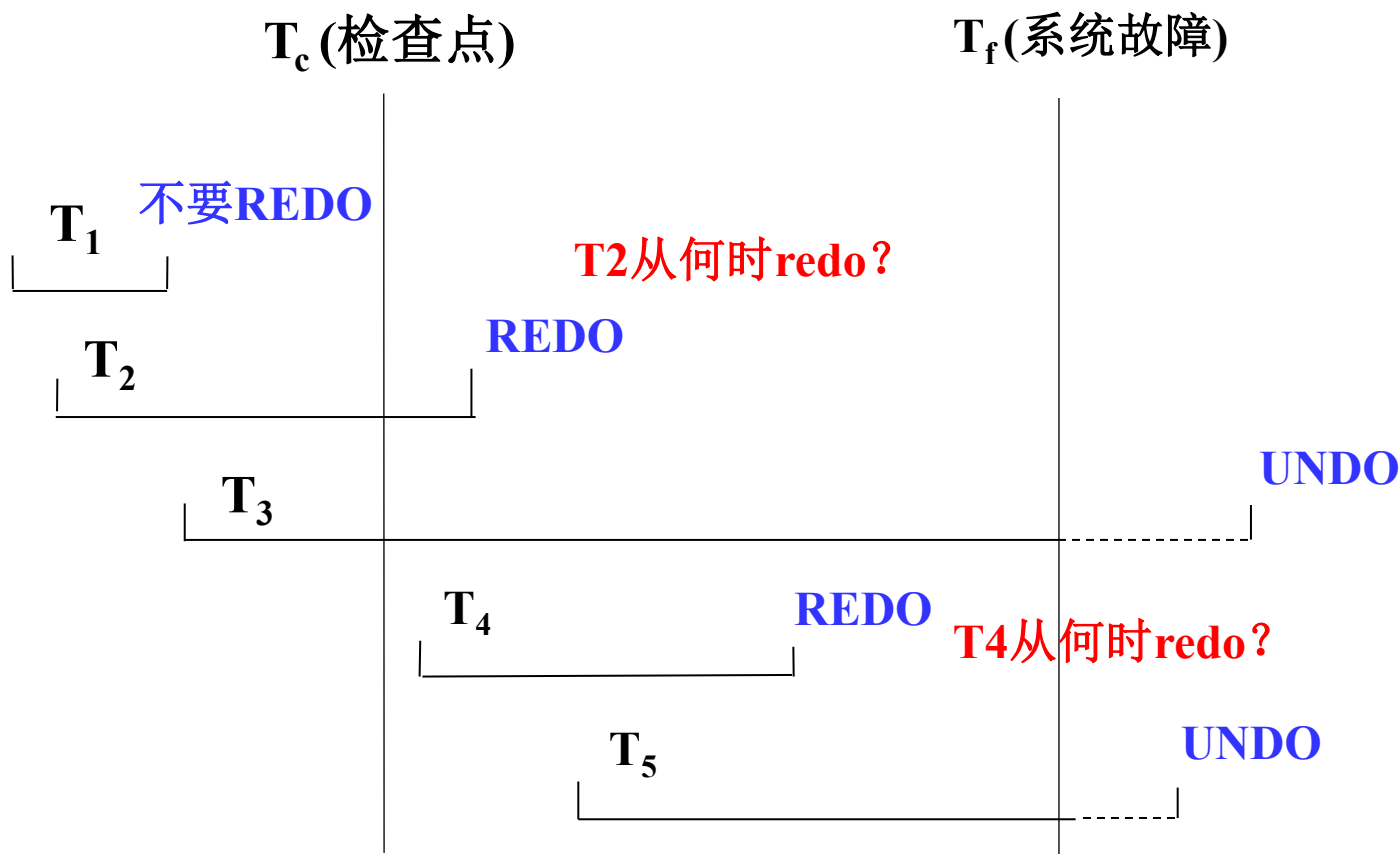
- 使用检查点方法可以改善恢复效率
 - 当事务T在一个检查点之前提交
 - T对数据库所做的修改已写入数据库
 - 写入时间是在这个检查点建立之前或在这个检查点建立之时
 - 在进行恢复处理时，没有必要对事务T执行REDO操作
 - 当事务T在检查点时未完成
 - T对DB的操作不确定是否写入数据库
 - 恢复处理时，如果需要REDO T，重做的起始点是检查点



利用检查点的恢复策略（续）

71

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）

72

- **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

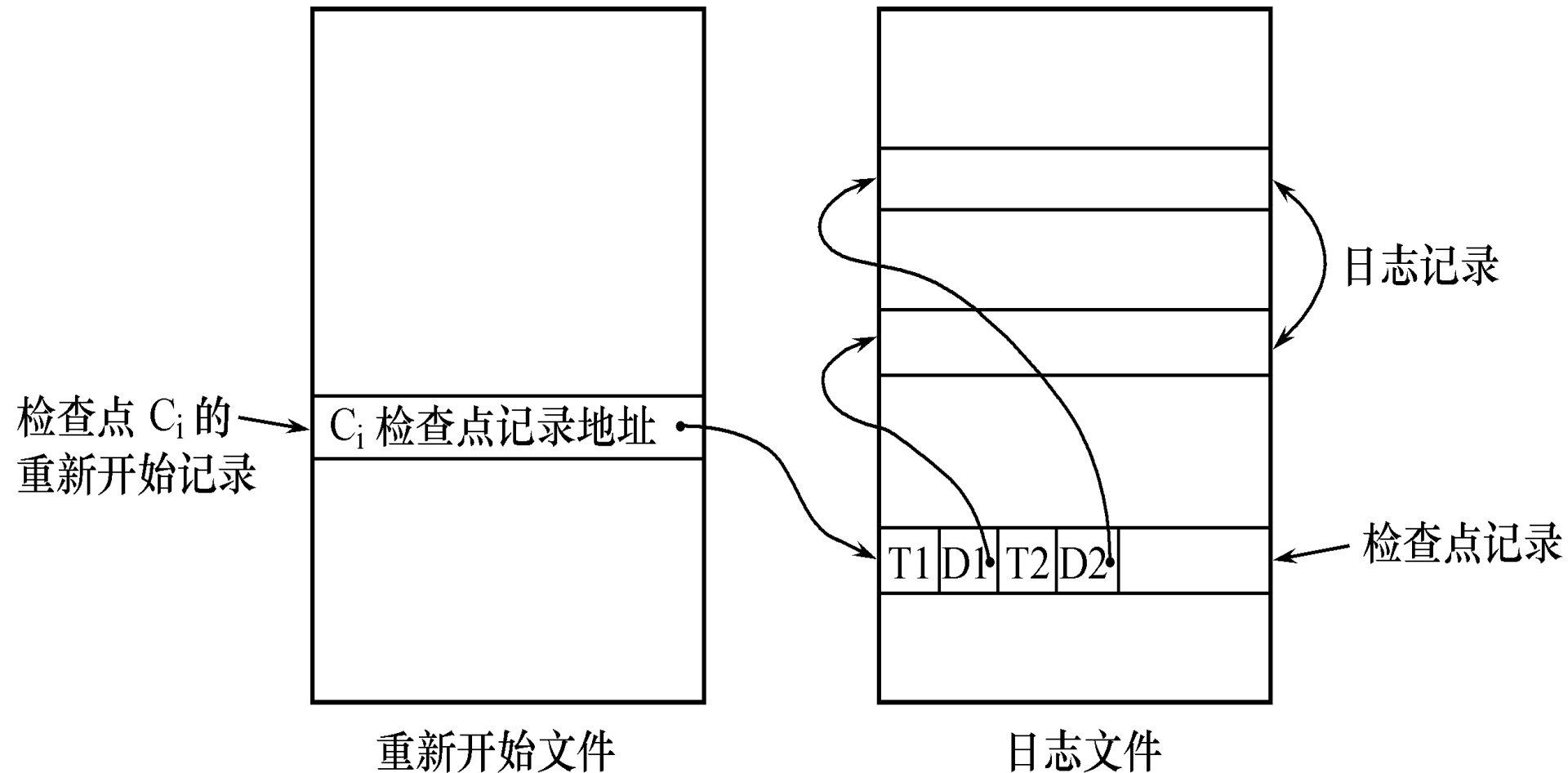
恢复策略：

- **T3和T5**在故障发生时还未完成，所以予以撤销**UNDO**
- **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

73



具有检查点的日志文件和重新开始文件



利用检查点的恢复步骤

74

- 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**
- 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）

75

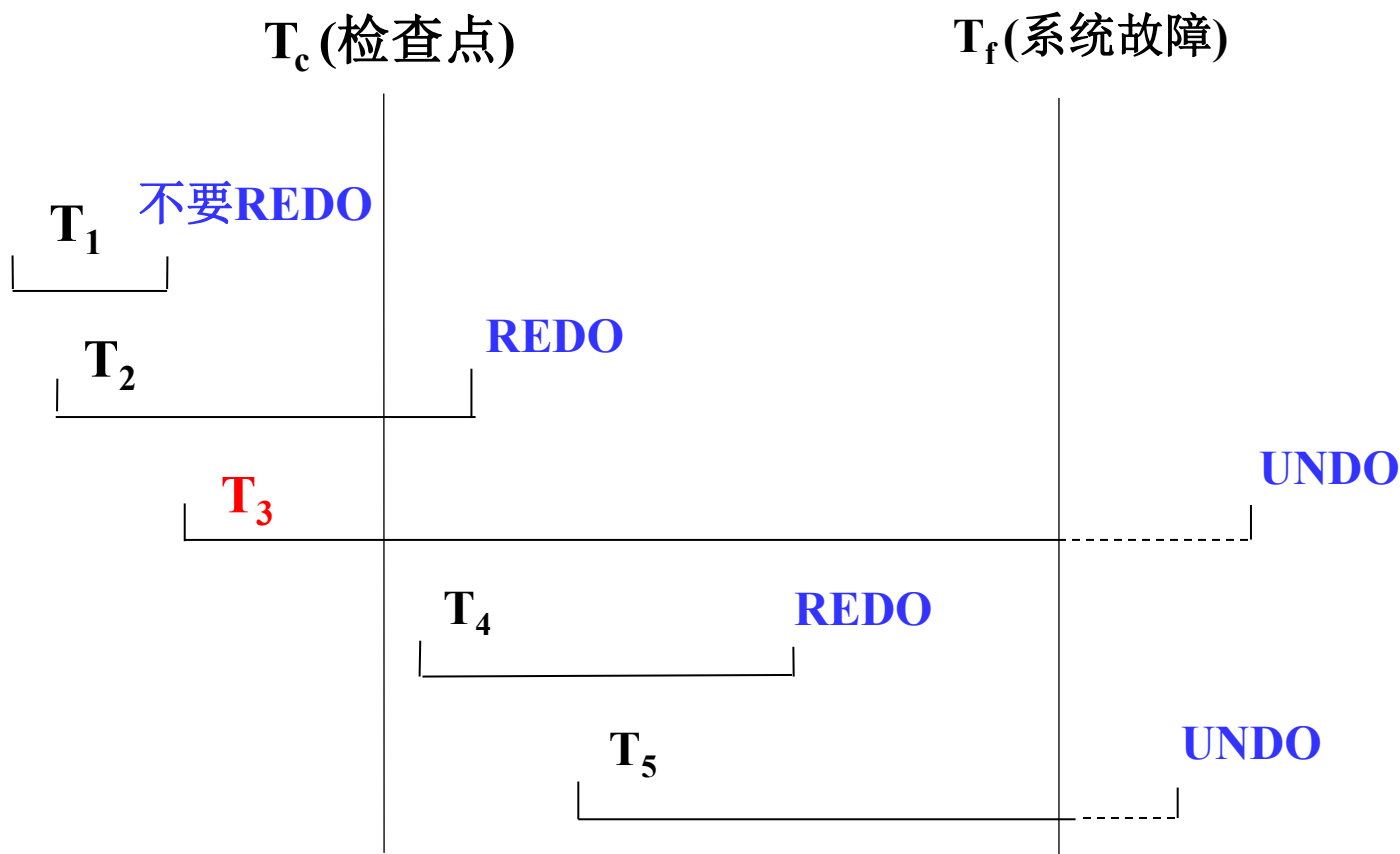
- 3.从检查点开始正向扫描日志文件，直到日志文件结束
 - 如有新开始的事务 T_i ，把 T_i 暂时放入UNDO-LIST队列
 - 如有提交的事务 T_j ，把 T_j 从UNDO-LIST队列移到REDO-LIST队列
- 4.反向扫：对UNDO-LIST中的每个事务执行UNDO操作
- 正向扫：对REDO-LIST中的每个事务执行REDO操作



利用检查点的恢复策略（续）

76

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





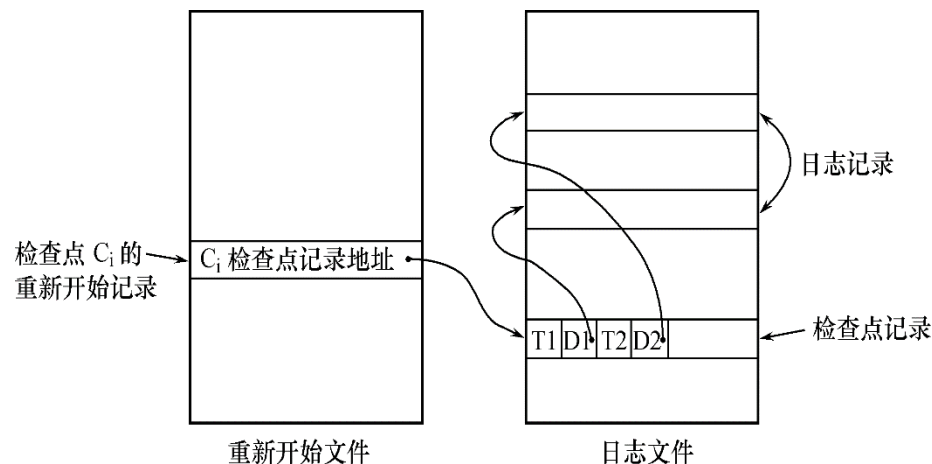
思考一：动态维护日志文件的方法

77

建立检查点

- **1.刷日志缓冲区：**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.记录检查点：**在日志文件中**写入一个检查点记录**
- **3.刷数据：**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.记重新开始文件：**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**

问，如果在2-3之间出故障了？
如果在3-4之间出故障了？





思考二：建立检查点的作用

78

□ 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（通常是DBMS自动做）

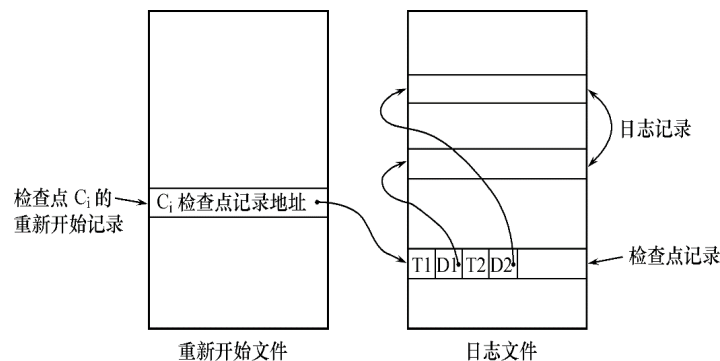
■ 定期

➤ 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（如何选择规则？）

➤ 按照某种规则，如日志文件已写满一半建立一个检查点

➤ 扔日志文件？怎么扔？





第十一章 数据库恢复技术

79

11.1 事务的基本概念

11.2 数据库恢复概述

11.3 故障的种类

11.4 恢复的实现技术

11.5 恢复策略

11.6 具有检查点的恢复技术

11.7 数据库镜像

11.8 小结



11.7 数据库镜像

80

- 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**）

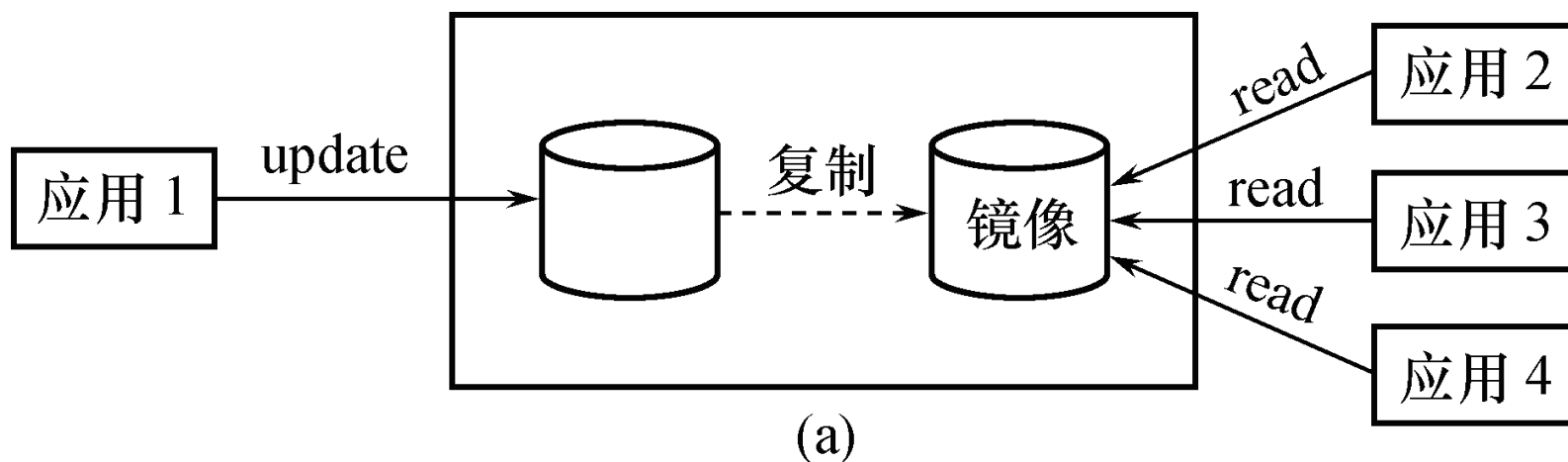


数据库镜像（续）

81

□ 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库的一致性
- 每当主数据库更新时，DBMS自动把更新后的数据复制过去

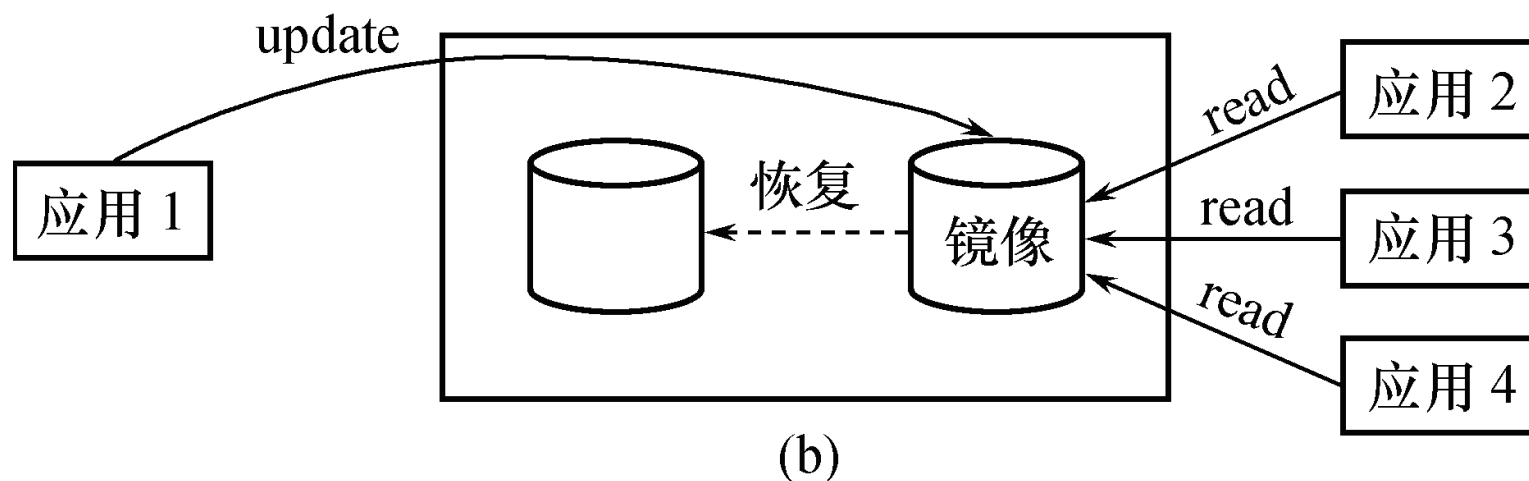




数据库镜像的用途

82

- 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
 - 不需要关闭系统和重装数据库副本(如下图所示)

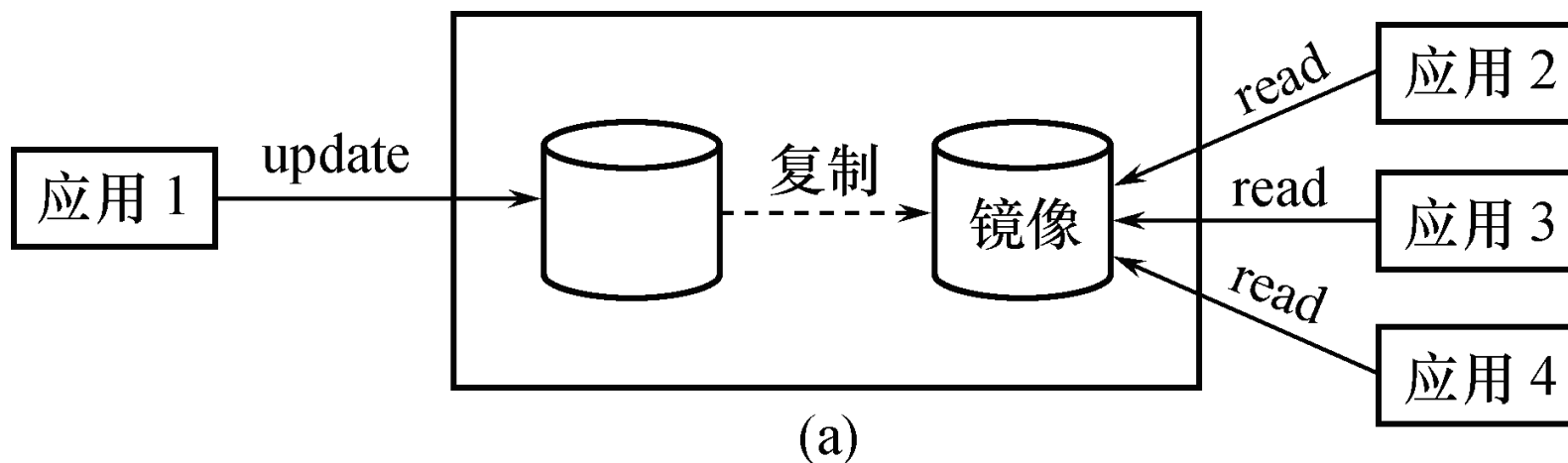




数据库镜像（续）

83

- 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，而不必等待该用户释放锁

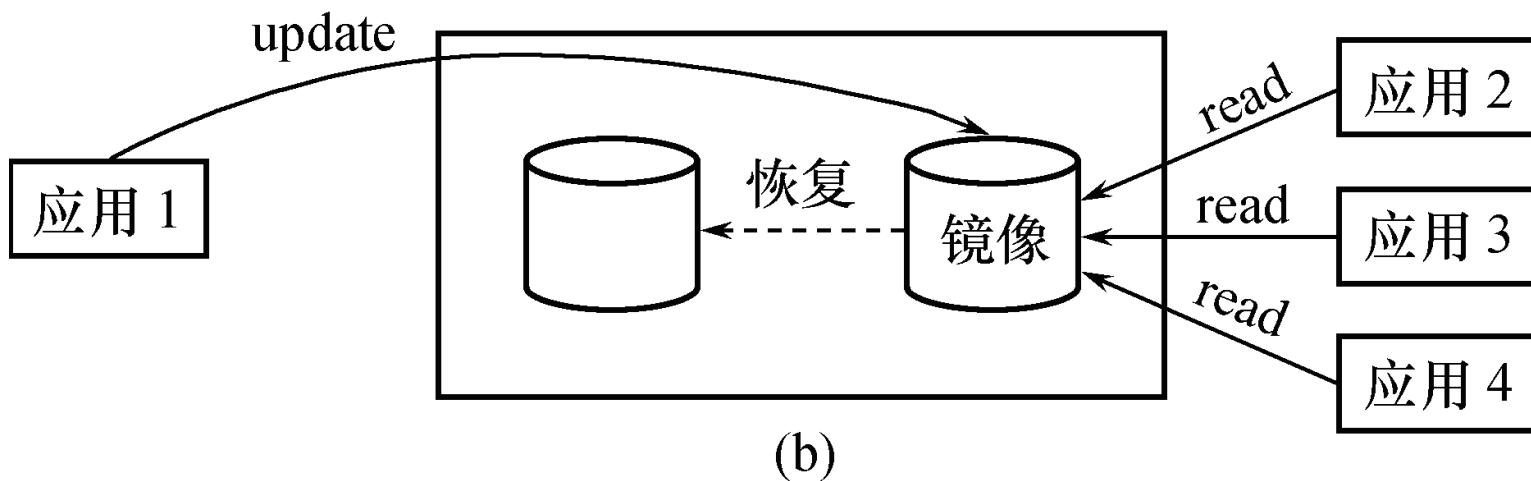




数据库镜像（续）

84

- 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，而不是对整个数据库进行镜像





第十一章 数据库恢复技术

85

11.1 事务的基本概念

11.2 数据库恢复概述

11.3 故障的种类

11.4 恢复的实现技术

11.5 恢复策略

11.6 具有检查点的恢复技术

11.7 数据库镜像

11.8 小结



11.8 小结

86

- 如果数据库只包含成功事务提交的结果，就说数据库处于一致性状态。保证数据一致性是对数据库的最基本的要求。
- 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）

87

- **DBMS**必须对事务故障、系统故障和介质故障进行恢复
- 恢复中最经常使用的技术：数据库转储和登记日志文件
- 恢复的基本原理：利用存储在后备副本、日志文件和数据库镜像中的冗余数据来重建数据库



小结（续）

88

□ 常用恢复技术

□ 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）

89

□ 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