



数据库系统原理

李瑞轩

华中科技大学计算机学院

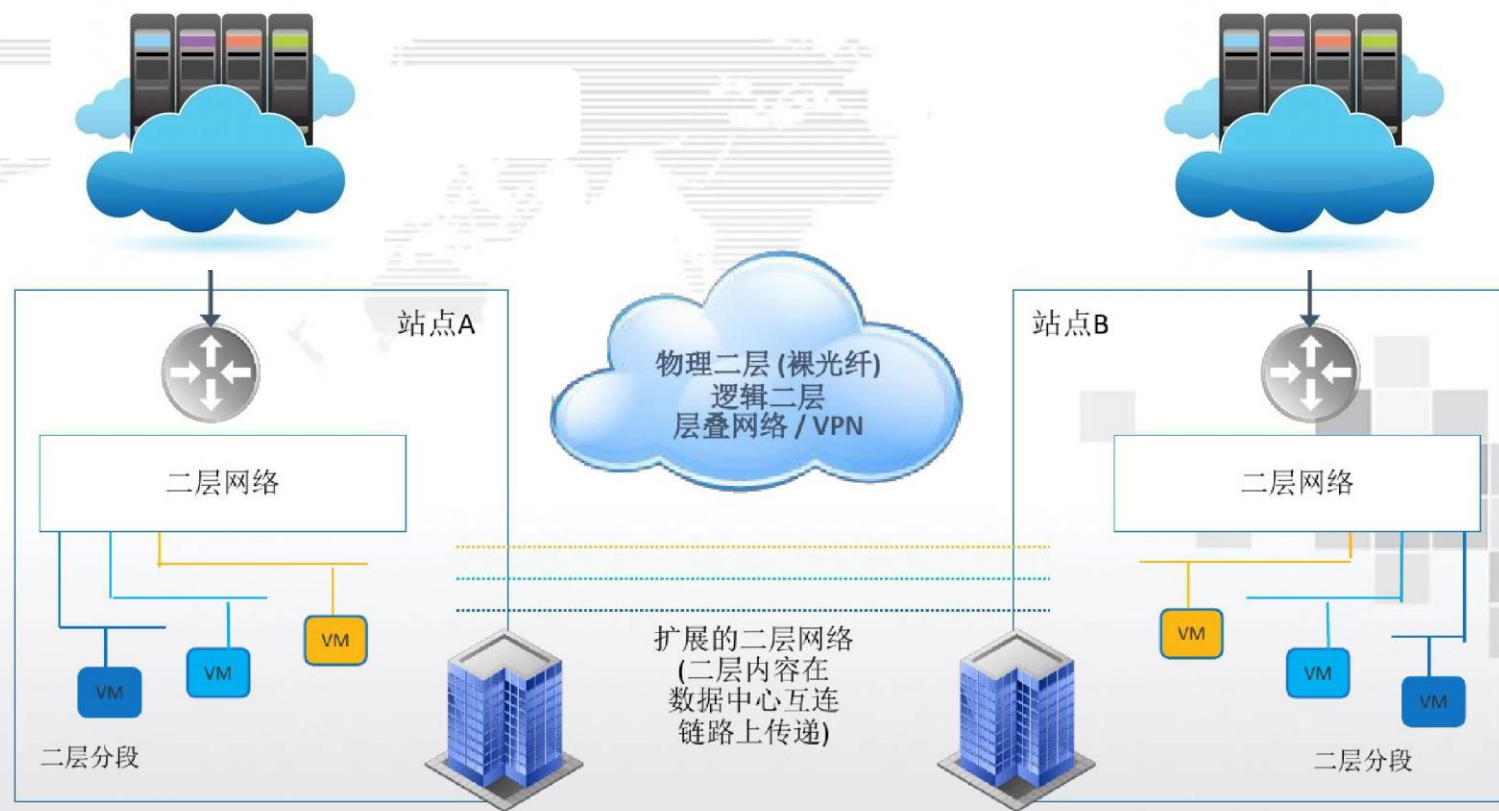


第十一章 数据库恢复技术

*I always say, keep a diary and someday it'll
keep you.*

---- Mae West

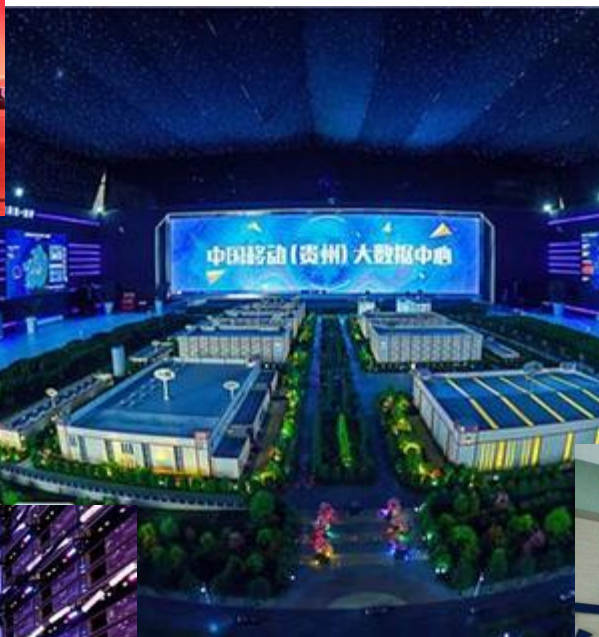
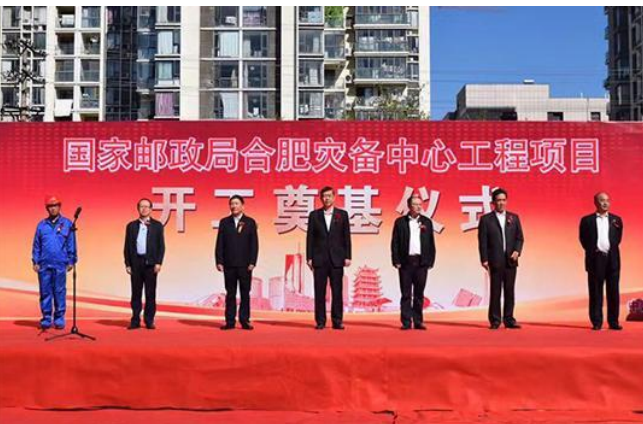
国家大数据灾备中心



15

国家大数据灾备中心由**人民网**联合国家**工信部**等有关中央部委共同成立，2018年9月在宁夏中卫市注册人民数据管理有限公司。

数据灾备技术已成为数据管理的核心需求





• 学习内容

- 11.1 事务的基本概念及其特性
- 11.2 数据库恢复的基本概念
- 11.3 数据库恢复的实现技术
- 11.4 数据库恢复的基本策略
- 11.5 数据库镜像的基本概念



• 学习目标

- 理解并掌握事务的基本概念及其特性
- 掌握数据库恢复的基本概念
- 掌握数据库恢复的基本实现技术
- 了解数据库恢复的基本策略

11.1 事务的基本概念及其特性

- 11.1.1 事务的基本概念
- 11.1.2 事务的ACID特性
- 11.1.3 事务的状态
- 11.1.4 事务ACID特性的实现

一笔电商交易的基本处理流程——事务

——天猫双十一中交易额——

00:00:26

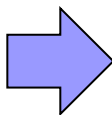
订单创建峰值达
58.3万笔/秒

11月1日至11日 00:30:00

3723亿

11月1日至11日 00:35:00

“亿元俱乐部”达到
342个



1

第一步，商业客户向销售商订货，首先要发出“用户订单”，该订单应包括产品名称、数量等等一系列有关产品问题。

2

第二步，销售商收到“用户订单”后，根据“用户订单”的要求向供货商查询产品情况，发出“订单查询”。

3

第三步，供货商在收到并审核完“订单查询”后，给销售商返回“订单查询”的回答。基本上是有无货物等情况。

4

第四步，销售商在确认供货商能够满足商业客户“用户订单”要求的情况下，向运输商发出有关货物运输情况的“运输查询”。

5

第五步，运输商在收到“运输查询”后，给销售商返回运输查询的回答。如：有无能力完成运输，及有关运输的日期、线路、方式等等要求。

6

第六步，在确认运输无问题后，销售商即刻给商业客户的“用户订单”一个满意的回答，同时要给供货商发出“发货通知”，并通知运输商运输。

7

第七步，运输商接到“运输通知”后开始发货。接着商业客户向支付网关发出“付款通知”。支付网关和银行结算票据等。

8

第八步，支付网关向销售商发出交易成功的“转账通知”。

11.1.1 事务的基本概念

■ 概念

- 用户定义的一个数据库操作序列
- 这些操作要么全做，要么全不做
- 一个不可分割的工作单位

■ 事务和程序

- **关系数据库事务**：一个/一组SQL语句，或一段程序。
- 一个程序通常包含多个事务

11.1.1 事务的基本概念

■ 定义事务

□ 显式定义方式

```
BEGIN TRANSACTION
```

```
SQL 语句1
```

```
SQL 语句1
```

```
.....
```

```
COMMIT / ROLLBACK
```

□ 隐式方式

当用户没有显式地定义事务时，DBMS按缺省规定自动划分事务。

11.1.1 事务的基本概念(续)

□ 示例

银行转帐：事务T从A帐户过户50 ￥ 到B帐户

```
T:   read(A);  
      A := A - 50;  
      write(A);  
      read(B);  
      B := B + 50;  
      write(B);
```

read(X): 从数据库传送数据项X到事务的工作区中

write(X): 从事务的工作区中将数据项X写回数据库

11.1.2 事务的ACID特性

■ 原子性 (Atomicity)

- 事务的所有操作在数据库中要么全部正确反映出来，要么全部不反映。（All-or-Nothing）

■ 一致性 (Consistency)

- 事务执行的结果必须是使数据库从一个一致性状态转变到另一个一致性状态。

一致性状态是指数据库中只包含成功事务提交的结果，没有夭折事务残留的修改。

11.1.2 事务的ACID特性(续)

■ 隔离性 (Isolation)

- 多个事务并发执行时，系统必须保证事务的执行不被其他事务干扰。每个事务都感觉不到系统中有其他事务在并发地执行。
- 对任何一对事务T1，T2，在T1看来，T2要么在T1开始之前已经结束，要么在T1完成之后再开始执行。

11.1.2 事务的ACID特性(续)

■ 隔离性示例

Begin Transaction

R(A)

A=A-1

W(A)

Commit;

Begin Transaction


R(A)

A=A-3

W(A)

Commit

| T1 | T2 |
|----------------------|-----------------|
| (1) R: A=16 | |
| (2) | R: A=16 |
| (3) A=A-1 写回 A=15 | |
| (4) | A=A-3 写回A=13 |



事务T1和T2没有保证隔离性

11.1.2 事务的ACID特性(续)

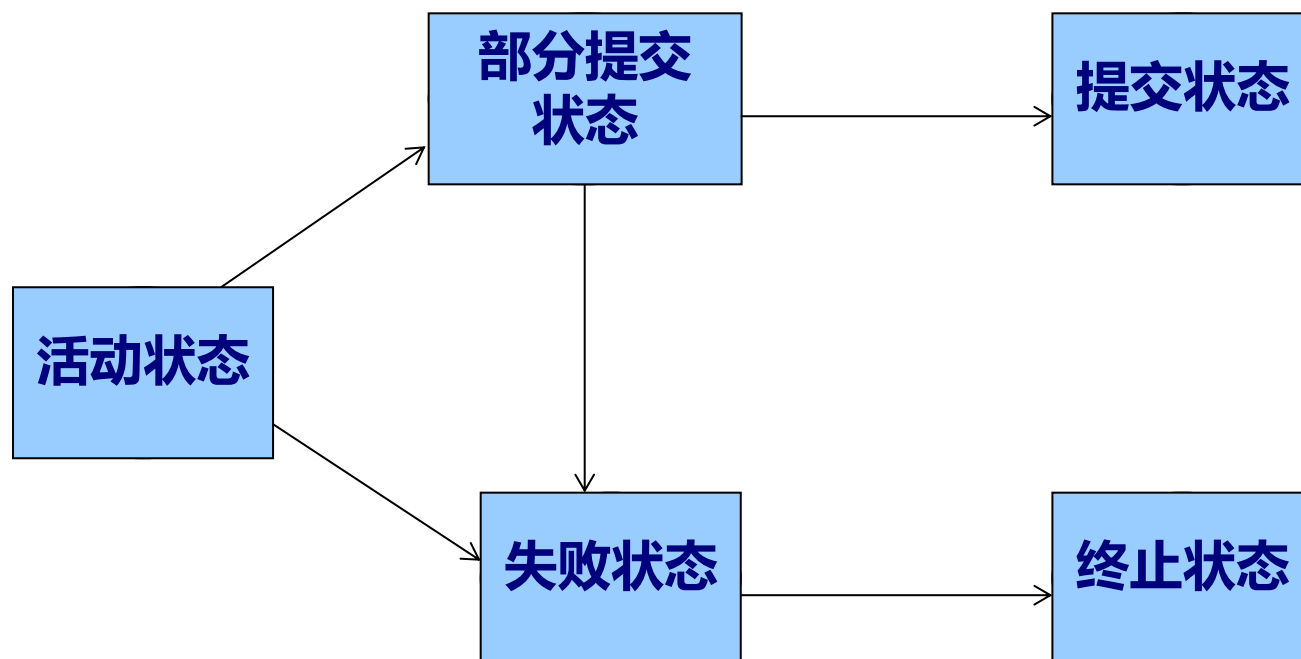
■ 持久性 (Durability)

- 一个事务成功完成后，它对数据库的改变必须是永久性的，即使系统可能出现故障。
 - 即使出现任何故障如断电等，事务一旦提交，则该提交产生的影响是永久的。
 - 一般通过先写日志（WAL）协议来保障。

11.1.3 事务的状态

- **活动状态**：事务执行时所处于的状态
- **部分提交状态**：最后一条语句被执行后
- **失败状态**：发现正常的执行不能继续后
- **终止状态**：事务回滚并且数据库已被恢复到事务开始执行前的状态后
- **提交状态**：事务成功完成后

事务的状态图例



提交的或终止的事务被称为**已经结束的事务**

11.1.4 事务ACID特性的实现

■ 原子性和持久性

- 由DBMS的恢复管理部件来实现，保证发生故障时一个事务对数据库的修改要么全部反映到数据库中，要么完全不反映。
 - 原子性和持久性由**恢复机制**实现。
- 早期使用的技术：**影子拷贝技术**
 - 效率低，不允许并发执行
- 当前采用的技术：**日志，备份**

11.1.4 事务ACID特性的实现(续)

■ 一致性

- 单个事务的一致性由应用程序员负责，完整性约束的自动检查对此提供支持
 - 一致性通过**并发控制机制**实现。

■ 隔离性

- 多个事务并发执行的正确性由DBMS的并发控制机制提供支持，通过对并发事务的合理调度来保证每个事务的一致性和隔离性
 - 隔离性通过**并发控制机制**实现。

一个数据库事务处理过程中可能出现各种故障

- 停/掉电
- 特定类型的硬件错误（CPU、硬盘故障等）
- 软件故障（DBMS, OS, APPS）
- 操作失误
- 造成系统停止运转的任何事件，使得系统要重启
-

停电！



11.2 数据库故障的种类

■ 事务内部的故障

- 有的可以通过事务程序本身发现的。如：转账事务、非法输入、找不到数据等。
- 有的是非预期的，不能由应用程序处理。如：运算溢出、死锁、违反完整性限制等。
- 特点：
 - 非预期事务故障：事务没有达到预期的终点（COMMIT或ROLLBACK），数据库可能处于不正确状态，DBMS仍在正常运行。
- 事务故障恢复：撤销事务 (UNDO)，DBMS自动完成。

11.2 数据库故障的种类

□ 事务内部的故障示例

```
Create table t1 (  
    a int,  
    b int check (b>2)  
)  
Begin trans  
    insert into t1 values(1,5)  
    insert into t1 values(2,0)  
Commit
```

```
SET XACT_ABORT  
{ ON | OFF }
```

- 当为**OFF**（默认）时：
只回滚产生错误的**SQL**语句，而事务将继续进行处理；
- 当为**ON**时，
如果**SQL**语句运行产生错误，整个事务将终止并回滚。

11.2 数据库故障的种类

■ 系统故障：软故障（Soft Crash）

- 造成系统停止运转的任何事件，使得系统要重新启动。
- 软件故障（DBMS, OS, APS）；
- 操作失误；
- 特定类型的硬件错误（CPU故障等）；
- 停/掉电。

■ 特点：

- 整个系统的正常运行突然被破坏，DBMS不能正常运行；
- 所有正在运行的事务都非正常终止；
- 内存数据丢失；外存数据不受影响；
- DB处于不正确或不一致状态：一些尚未完成事务的结果可能已送入DB；已完成事务的结果可能部分还未送入DB；已完成事务的结果全部未送入DB（未及提交）。

11.2 数据库故障的种类

■ 系统故障恢复

- ✓ 发生系统故障时，事务未提交：
 - 恢复策略：强行撤消（UNDO）所有未完成事务
- ✓ 发生系统故障时，事务已提交，但缓冲区中的信息尚未完全写回到磁盘上：
 - 恢复策略：重做（REDO）所有已提交的事务

11.2 数据库故障的种类

- **介质故障**：磁盘损坏、磁头碰撞、强磁场干扰等外存故障
- **特点**：
 - 数据库遭到破坏，存在外存的数据部分丢失或全部丢失，正在执行的事务中断。
 - 发生可能性小
 - 破坏性最大
- **介质故障恢复**：
 - **装入**数据库发生介质故障前某个时刻的数据**副本**
 - **重做REDO**自此时始的所有**成功事务**，将这些事务已提交的结果重新记入数据库

11.2 数据库故障的种类

■ 计算机病毒

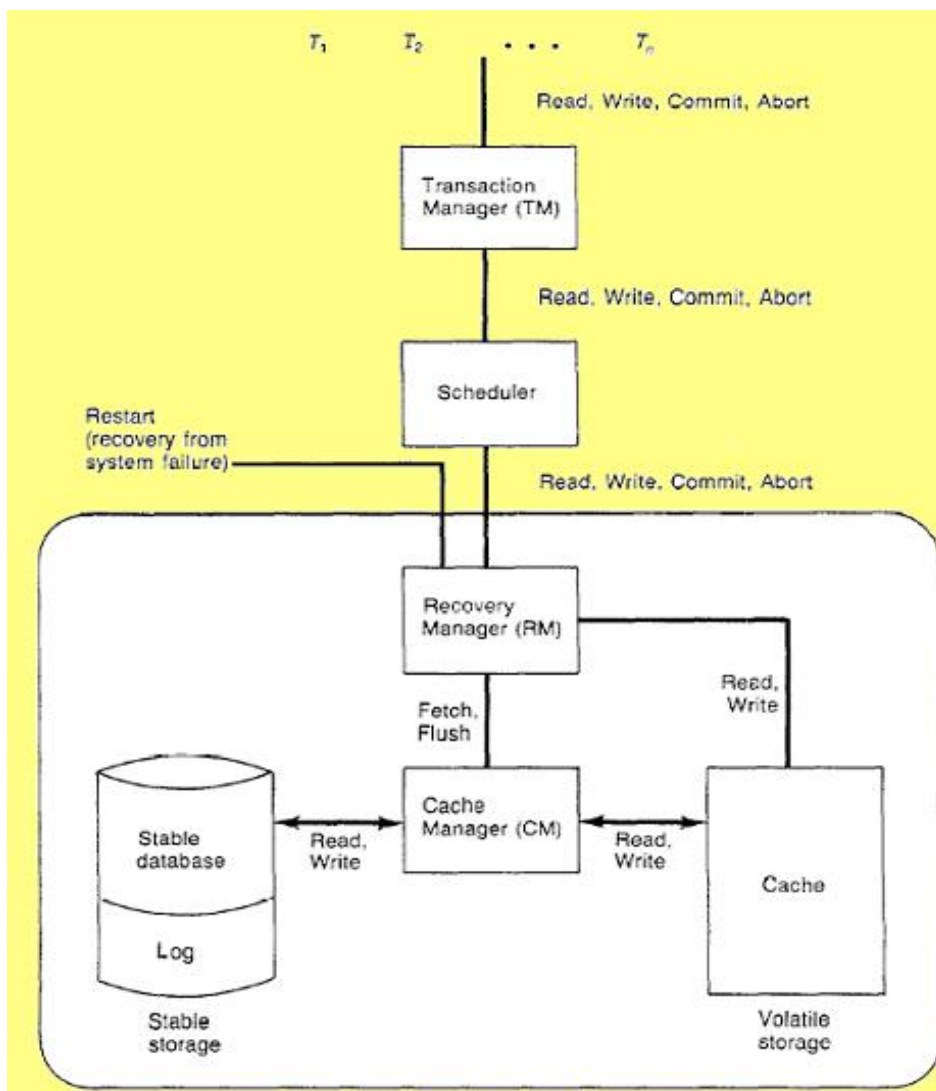
- 一种人为的故障或破坏，是一些恶作剧者研制的一种计算机程序
- 可以繁殖和传播

■ 危害

- 破坏、盗窃系统中的数据
- 破坏系统文件
- 数据库本身被破坏，或者DB正常，但数据可能不正确

11.3 数据库恢复的实现技术

- 数据库恢复部件，
在数据库管理系统
中的位置



11.3 数据库恢复的实现技术

■ 核心思想

□ 冗余

- 利用冗余副本，重建数据库，使其达到一致状态

■ 恢复技术涉及的关键问题

1. 如何建立数据的冗余副本？

- 数据转储 (backup)
- 登录日志文件 (logging)

2. 如何利用这些冗余副本实施数据库恢复？

11.3.1 数据转储

■ 1. 定义

- DBA定期地将整个数据库复制到磁带或另一个磁盘上保存起来的过程。
- 副本

■ 2. 实现

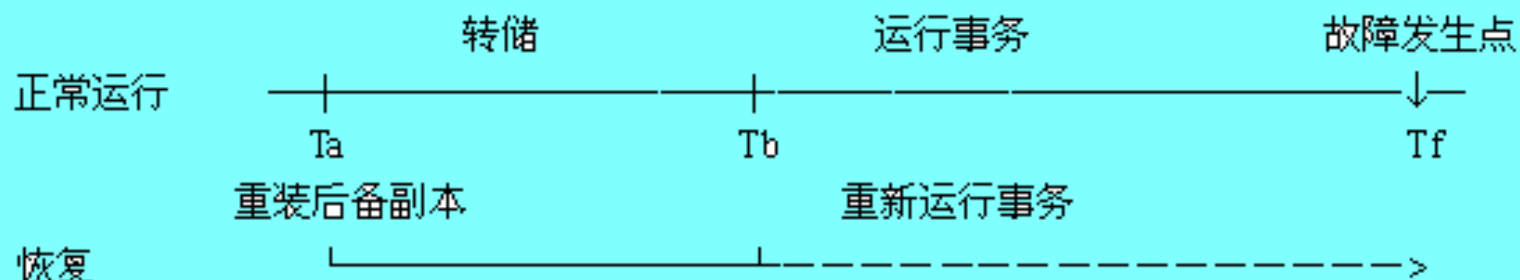


图 7.1 转储和恢复

11.3.1 数据转储(续)

■ 3. 分类

□ 转储时机

- 静态转储：无事务执行
- 动态转储：边执行事务，边转储

□ 转储内容

- 海量转储：转储全部数据库
- 增量转储：只转储更新过的数据

11.3.2 日志

■ 1. 定义

- 日志文件是用来记录事务对数据库的更新操作的文件

■ 2. 格式

- 以记录为单位
- 以数据块为单位

11.3.2 日志(续)

■ 3. 内容

□ 以记录为单位的日志

- 事务标识
- 操作的类型
- 操作对象
- 更新前数据的旧值
- 更新后数据的新值

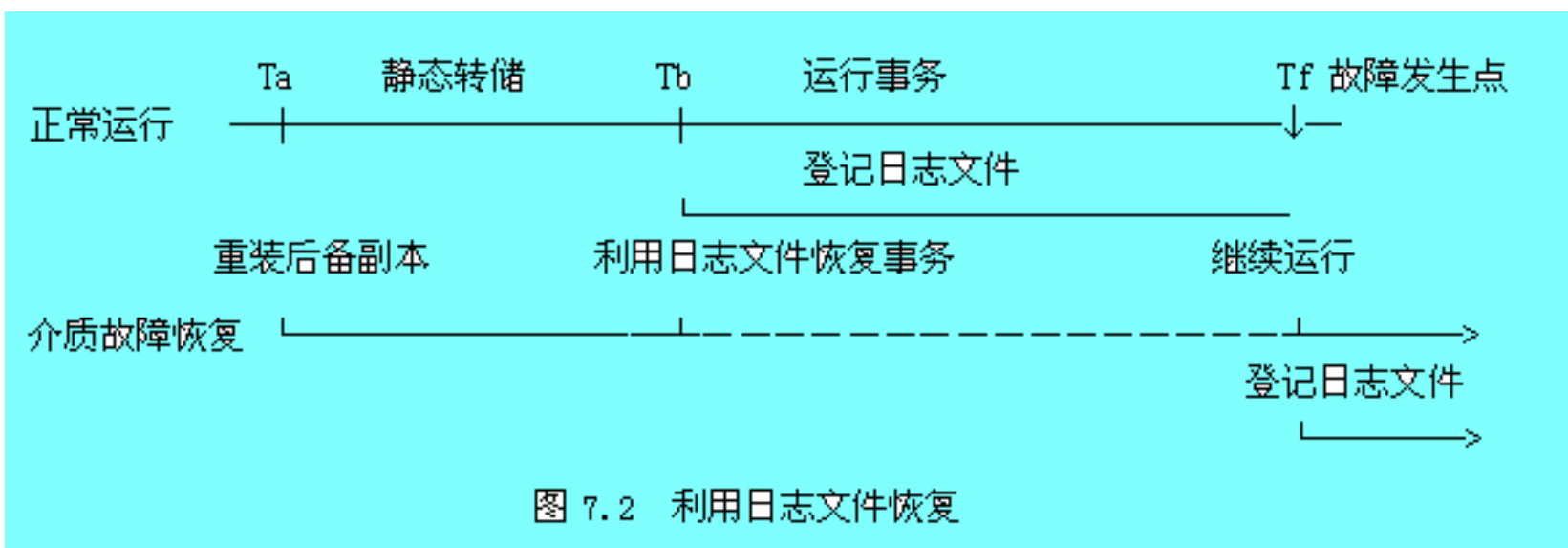
□ 以块为单位的日志

- 事务标识
- 更新前、后的块的内容

11.3.2 日志(续)

■ 4. 作用

- 事务故障恢复和系统故障恢复
- 协助后备副本进行介质故障恢复



11.3.2 日志(续)

■ 5. 记录日志文件

□ 日志缓冲区

- 在内存中开辟的临时保存日志记录的区域
- 根据需要一次将一个或多个缓冲块写入磁盘，从而减少写磁盘的次数。
- 写到磁盘中的日志记录顺序必须与写入日志缓冲区的顺序完全一致。

11.3.2 日志(续)

■ 5. 记录日志文件

□ 记录时机

- 必须先写日志文件，后写数据库
- Write-Ahead Log (WAL) Protocol

11.4 数据库恢复的基本策略

■ 1. 两种日志

- UNDO日志
- REDO日志

■ 2. 使用日志文件进行恢复

- ARIES算法（应用）（WAL + UNDO + REDO）
- 模拟算法（理论）

数据库事务及其日志文件示例

| Operation | Log file | |
|-----------|-----------------|------------|
| BEGIN T1 | (Begin T1) | |
| R1(A,50) | | 初始值:A 50 |
| W1(A,20) | (W,T1,A,50,20) | B 50 |
| BEGIN T2 | (Begin T2) | C 100 |
| R2(C,100) | | |
| W2(C,50) | (W,T2,C,100,50) | |
| Commit T2 | (Commit T2) | Recovery值: |
| R1(B,50) | | A 50 |
| W1(B,80) | (W,T1,B,50,80) | B 50 |
| <hr/> | | |
| Commit T1 | (Commit T1) | C 50 |

11.4 数据库恢复的基本策略(续)

■ 3. 使用日志文件进行恢复的思想 (ARIES)

- 从后往前扫描日志，构造undo-list和redo-list：
 - 对每一个形如<Ti commit>的记录，将Ti 加入redo-list。
 - 对每一个形如<Ti BEGIN>的记录，如果Ti不属于redo-list，则将Ti加入undo-list。
- 由后至前重新扫描日志，对undo-list中的每个事务Ti的每一个日志记录执行undo操作。
- 由前至后重新扫描日志，并且对redo-list中每个事务Ti的每一个日志记录执行redo操作。

11.4 数据库恢复的基本策略(续)

4. 事务故障的恢复 (UNDO)

- ①反向扫描日志文件 (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。
- 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。

11.4 数据库恢复的基本策略(续)

5. 系统故障的恢复 (UNDO + REDO)

恢复操作就是要撤销故障发生时未完成的事务，重做已完成的事务。

- ①正向扫描日志文件，找出在故障发生前已经提交事务，将其事务标识记入重做(REDO)队列。同时还要找出故障发生时尚未完成的事务，将其事务标识记入撤销(UNDO)队列。
- ②反向扫描日志文件，对撤销队列中的各个事务进行撤销(UNDO)处理。
- ③正向扫描日志文件，对重做队列中的各个事务进行重做(REDO)处理。

11.4 数据库恢复的基本策略(续)

6. 介质故障的恢复 (REDO)

- ①装入最新的后备数据库副本，使数据库恢复到最近一次转储时的一致性状态。

对于动态转储的数据库副本，还须同时装入转储开始时刻的日志文件副本，利用与恢复系统故障相同的方法(即 REDO+UNDO)，才能将数据库恢复到一致性状态。

- 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。

11.5 具有检查点的恢复技术

■ 1. 日志文件的缺点

- 耗费大量时间
- 重复执行
- 耗费大量空间

11.5 具有检查点的恢复技术(续)

■ 2. CHECK POINT技术

□ 基本策略

- 周期性地对日志做检查点，以避免故障恢复时检查整个日志。

□ 方法

- 在日志文件中增加一类新的记录——检查点(Checkpoint)记录，增加一个重新开始文件，并让恢复子系统在登录日志文件期间动态地维护日志。

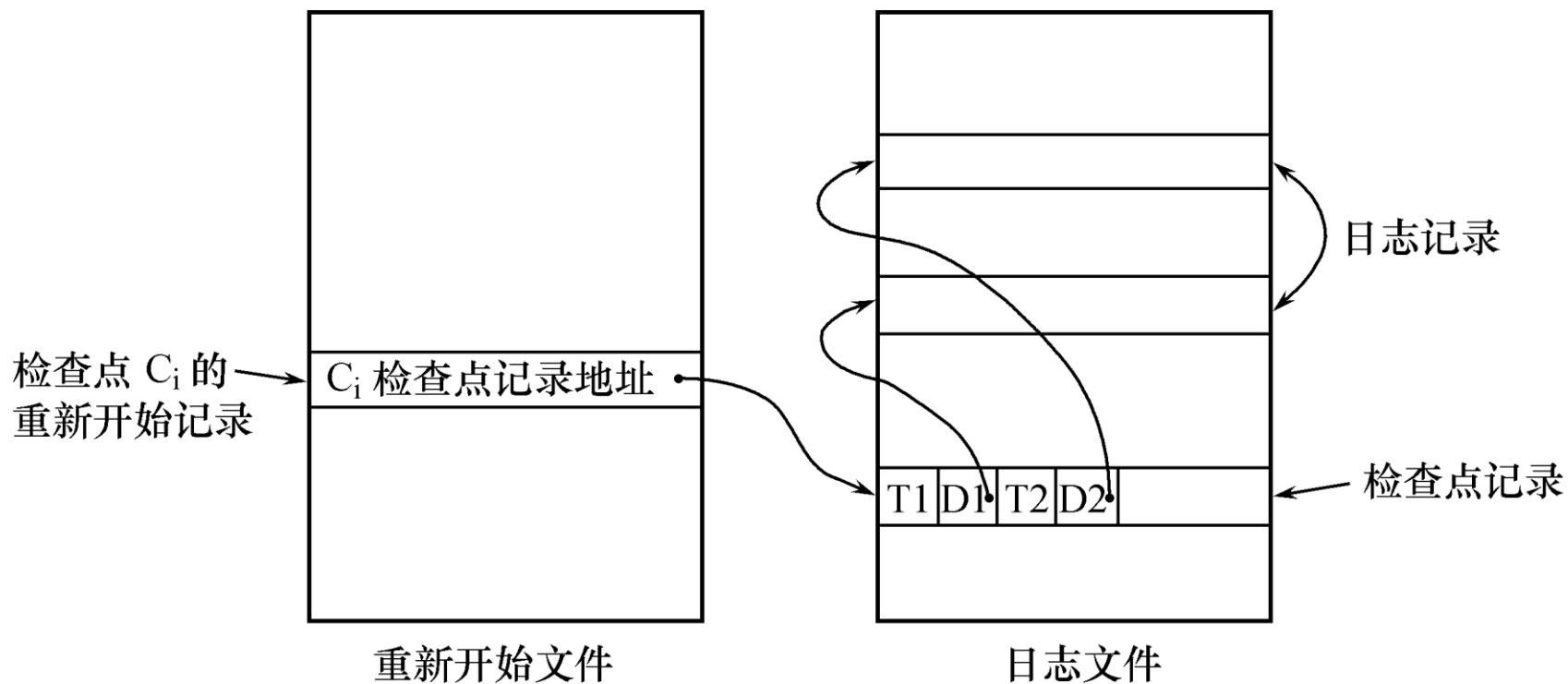
11.5 具有检查点的恢复技术(续)

■ 2. CHECK POINT技术

□ Checkpoint记录内容

- Checkpoint时刻所有正在执行的事务清单
- 这些事务最近一个日志记录的地址

11.5 具有检查点的恢复技术(续)



具有检查点的日志文件和重新开始文件

11.5 具有检查点的恢复技术(续)

■ 3. 动态维护日志文件的方法

- 将当前日志缓冲中的所有日志记录写入磁盘的日志文件上。
- 在日志文件中写入一个检查点记录。
- 将当前数据缓冲的所有数据记录写入磁盘的数据库中。
- 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。

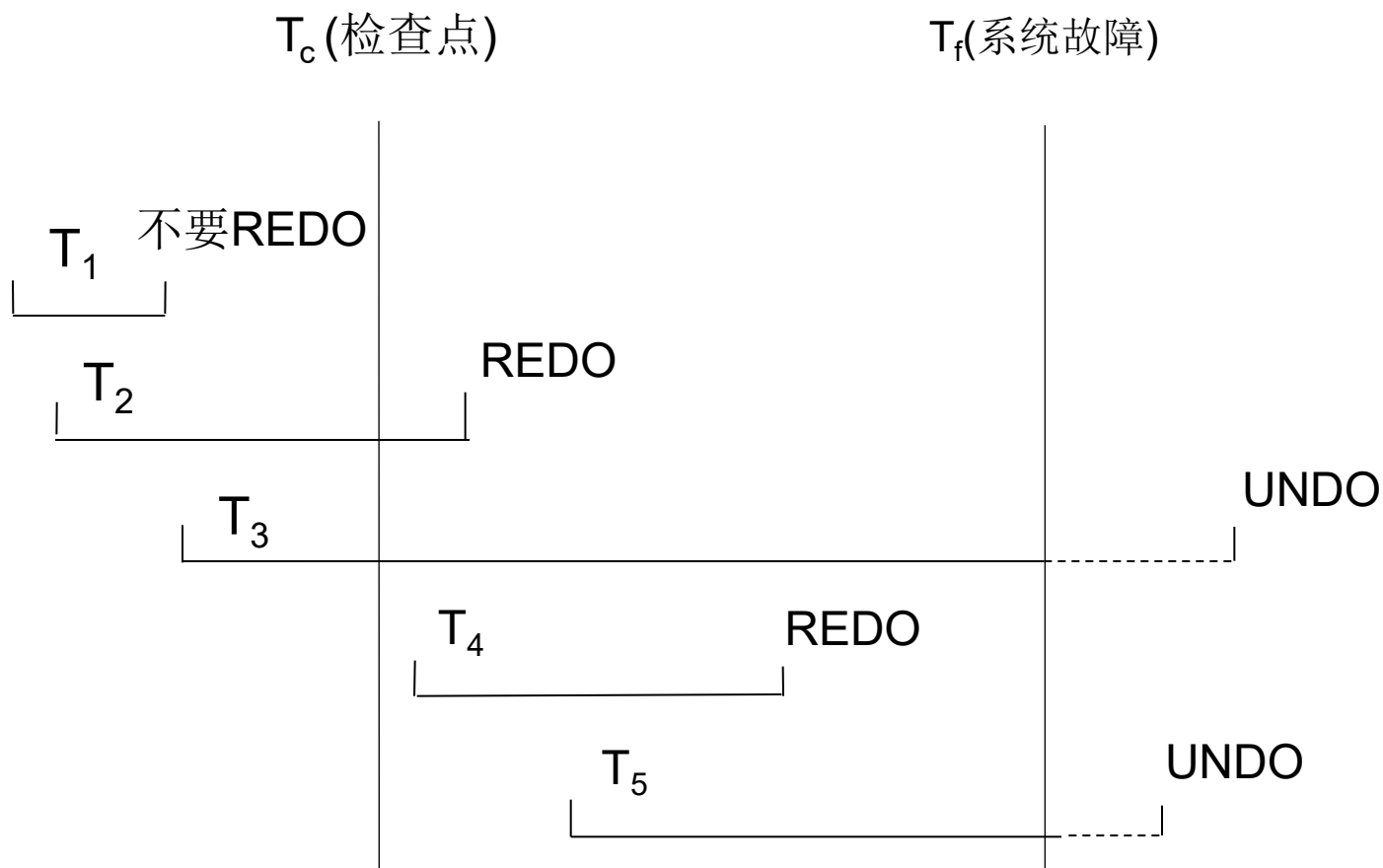
11.5 具有检查点的恢复技术(续)

■ 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

11.5 具有检查点的恢复技术(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



11.5 具有检查点的恢复技术(续)



11.5 具有检查点的恢复技术(续)

■ 4. 使用检查点方法进行恢复的步骤

- 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。
- 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。

11.5 具有检查点的恢复技术(续)

■ 4. 使用检查点方法进行恢复的步骤（续）

- 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
- 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

11.6 数据库镜像

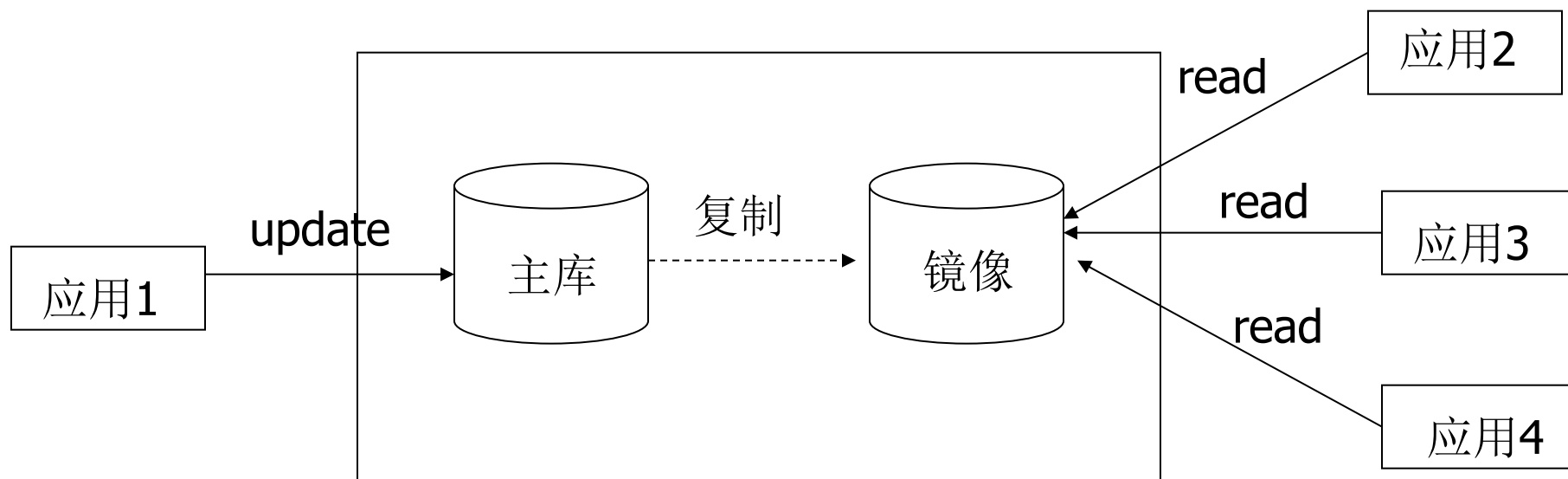
■ 概念

- 根据DBA的要求，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。每当主数据库更新时，DBMS自动把更新后的数据复制过去，即DBMS自动保证镜像数据与主数据的一致性。

■ 实现

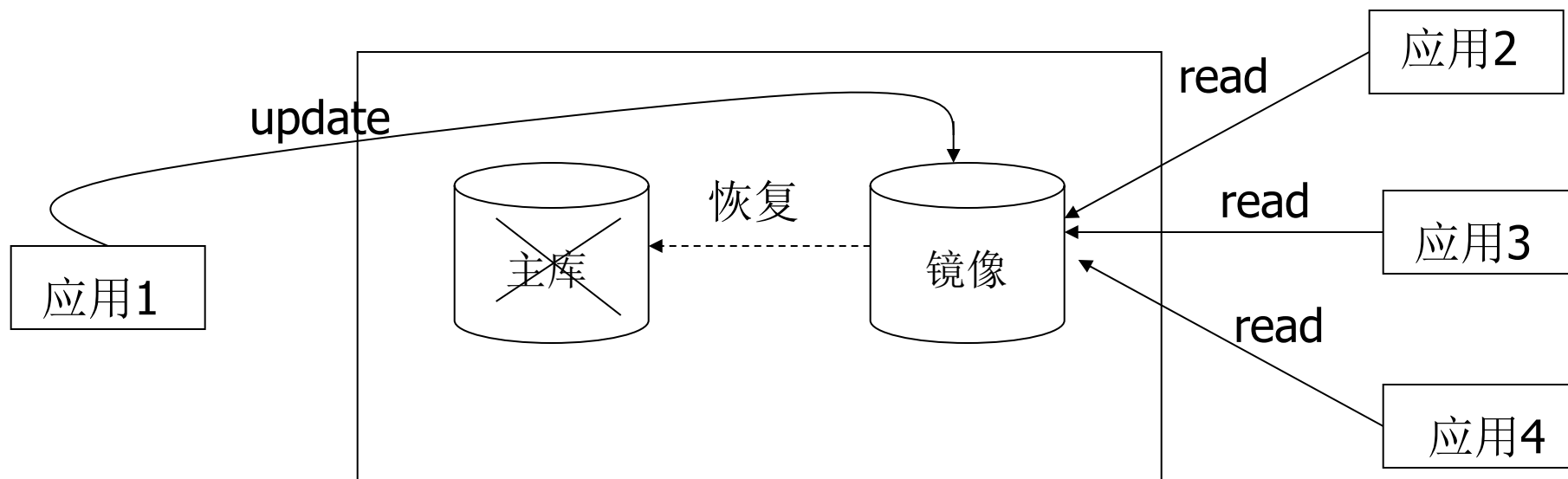
- 镜像关键数据和日志文件。

11.6 数据库镜像 (续)



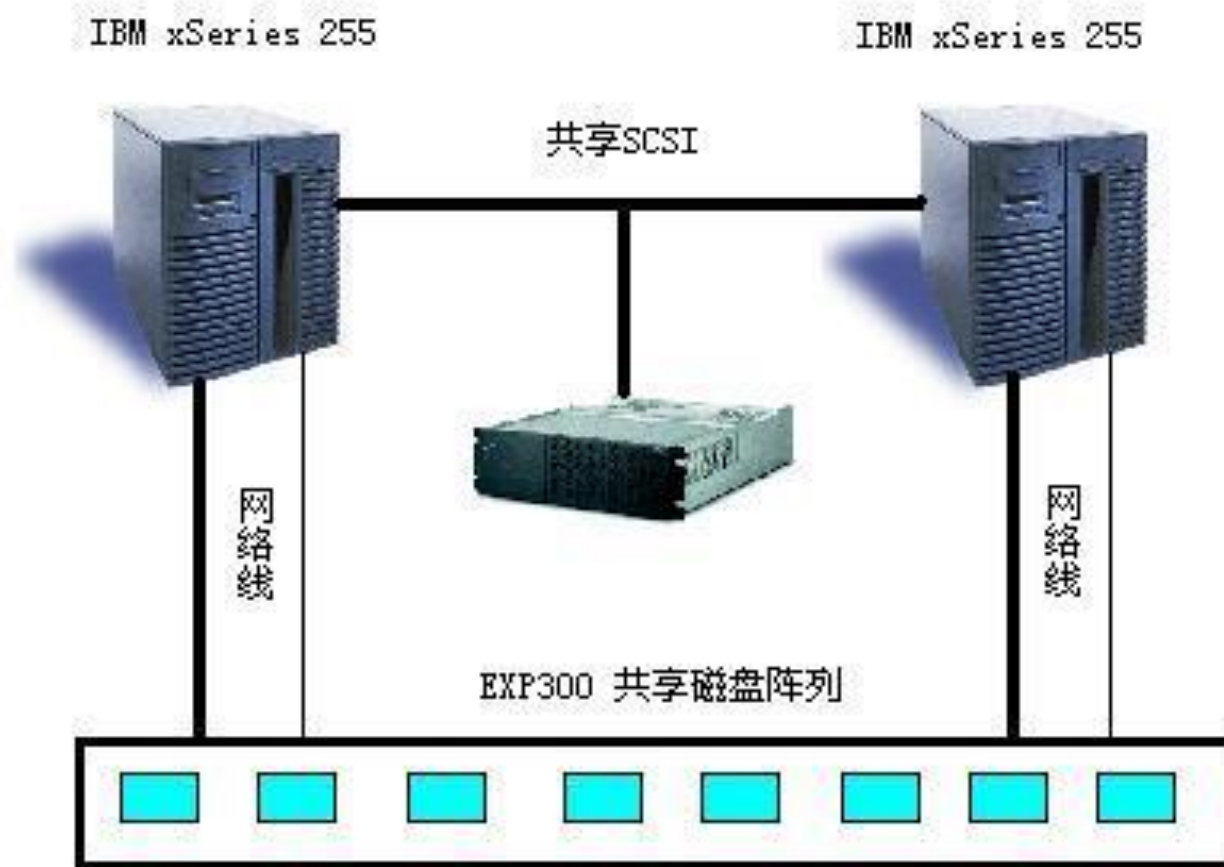
(a)

11.6 数据库镜像 (续)



(b)

双机热备示例



课堂练习

- **Q1:** 为什么事务非正常结束时会影响数据库数据的正确性？请举例说明。
- **Q2:** _____ 操作是反向扫描日志文件，撤销对数据库的更新操作，使DB恢复到更新前的状态；
_____ 操作是正向扫描日志文件，重新做一次更新，使DB恢复到更新后的状态；
- **Q3:** 请简述记录日志的方式的2大原则。

练习

- **Q4.**数据库系统发生故障时，可以基于日志进行恢复。下面列出的条目中，哪些是日志记录的内容_____。
 - I. 事务开始信息
 - II. 更新信息
 - III. 提交信息
 - IV. 事务终止信息

A) I、II和IV B) I、III和IV C) II、III和IV D) 都是
- **Q5.**试述WAL协议的必要性

下课了。。。。

探
索



休息一会儿。。。。

