

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



10.4 恢复的实现技术

恢复机制涉及的关键问题

1. 如何建立冗余数据

- 数据转储 (**backup**)

- 登记日志文件 (**logging**)

2. 如何利用这些冗余数据实施数据库恢复



10.4 恢复的实现技术

10.4.1 数据转储

10.4.2 登记日志文件



10.4.1 数据转储

1.什么是数据转储

2.转储方法



1.什么是数据转储

- ❖ 转储是指数据库管理员定期地将整个数据库复制到磁带、磁盘或其他存储介质上保存起来的过程
- ❖ 备用的数据文本称为**后备副本(backup)或后援副本**



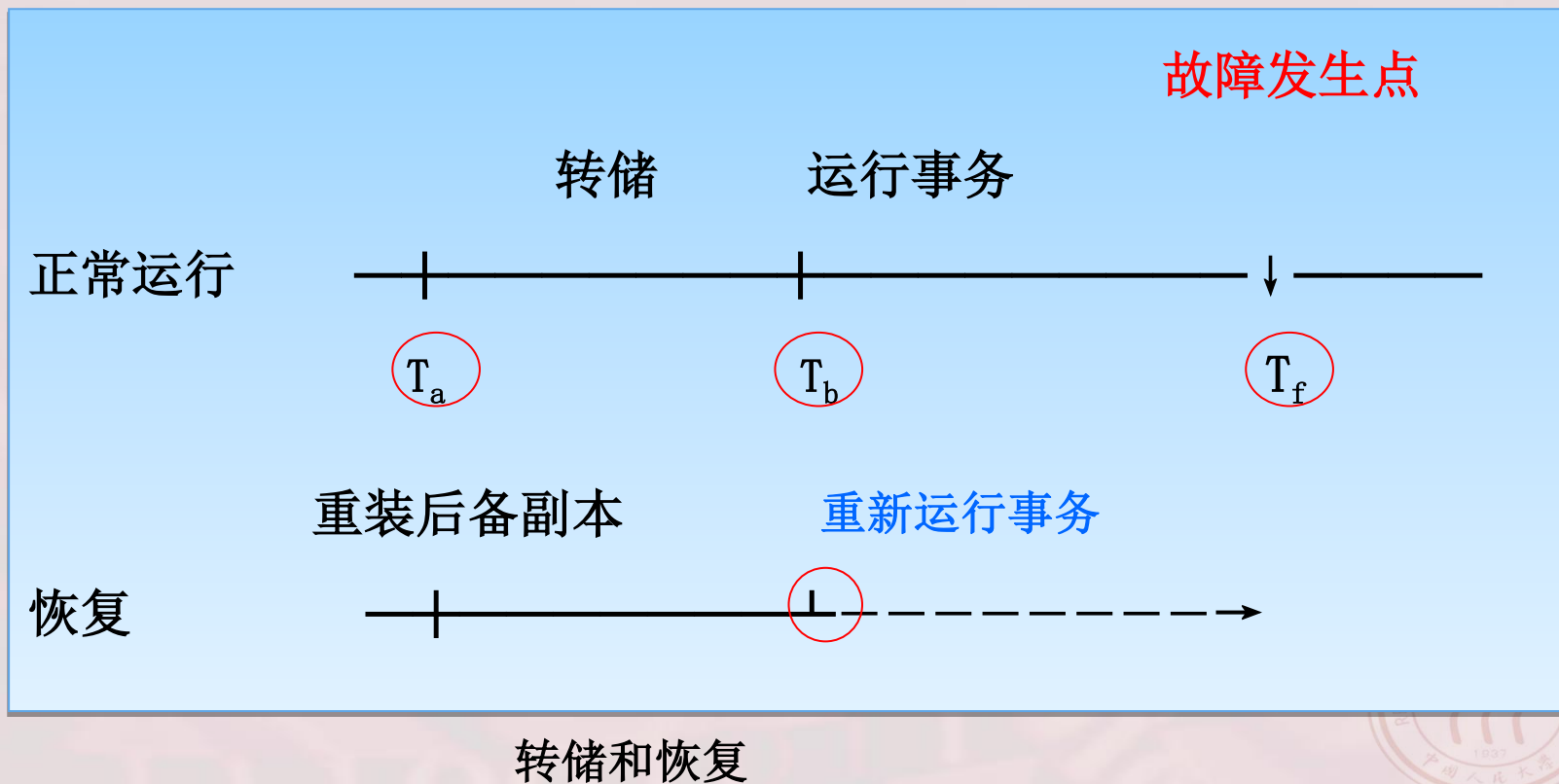
数据转储（续）

- ❖ 数据库遭到破坏后可以将后备副本重新装入
- ❖ 重装后备副本只能将数据库恢复到转储时的状态
- ❖ 要想恢复到故障发生时的状态，必须重新运行自转储以后的所有更新事务



数据转储（续）

[例]



2.转储方法

- (1) 静态转储与动态转储
- (2) 海量转储与增量转储
- (3) 转储方法小结



(1) 静态转储与动态转储

❖ 静态转储

- 在系统中无运行事务时进行的转储操作
- 转储开始时数据库处于一致性状态
- 转储期间不允许对数据库的任何存取、修改活动
- 得到的一定是一个数据一致性的副本
- 优点：简单
- 缺点：降低了数据库的可用性
 - 转储必须等待正运行的用户事务结束
 - 新的事务必须等转储结束



静态转储与动态转储（续）

❖ 动态转储

- 转储操作与用户事务并发进行
- 转储期间允许对数据库进行存取或修改
- 优点
 - 不用等待正在运行的用户事务结束
 - 不会影响新事务的运行
- 缺点
 - 不能保证副本中的数据正确有效



例:

转储开始

数据库

A	B	C	D
1	2	3	4
1	2	3	4



例:

转储开始

备份

数据库

A	B	C	D
1	2	3	4
1	2	3	4

T1	T2	备份
		副本A=1

转储过程

例:

转储开始

备份

数据库

A	B	C	D
1	2	3	4
1	7	3	4

T1	T2	备份
B:=7		副本A=1

转储过程

数据库

A	B	C	D
1	2	3	4
1	7		
1	7	3	4

T1	T2	备份
B:=7		副本A=1 副本B=7

转储过程

例:

转储开始

备份

数据库

A	B	C	D
1	2	3	4
1	7		
1	7	6	4

T1	T2	备份
B:=7	C:=6	副本A=1
		副本B=7

转储过程

例:

转储开始

备份

数据库

A	B	C	D
1	2	3	4
1	7	6	
1	7	6	4

T1	T2	备份
B:=7	C:=6	副本A=1
		副本B=2
		副本C=6

转储过程

例:

转储开始

备份

数据库

A	B	C	D
1	2	3	4
1	7	6	
5	7	6	4

T1	T2	备份
B:=7		副本A=1
	C:=6	副本B=2
A:=5		副本C=6

转储过程

例:

转储开始

备份

数据库

A	B	C	D
1	2	3	4
1	7	6	4
5	7	6	4

T1	T2	备份
B:=7 A:=5	C:=6	副本A=1
		副本B=2
		副本C=6
		副本D=4

转储过程

静态转储与动态转储（续）

❖ 利用动态转储得到的副本进行故障恢复

- 需要把动态转储期间各事务对数据库的修改活动登记下来，建立日志文件
- 后备副本加上日志文件就能把数据库恢复到某一时刻的正确状态



例:

转储开始

备份

数据库

A	B	C	D
1	2	3	4
1	7	6	4
5	7	6	4

T1	T2	备份
B:=7 A:=5 ROLLBACK	C:=6 commit	副本A=1
		副本B=2
		副本C=6
		副本D=4
		转储结束

转储过程

例:

转储开始

备份

数据库

A	B	C	D
1	2	3	4
1	7	6	4

T1	T2	备份
B:=7		副本A=1
		副本B=2
	C:=6	副本C=6
A:=5		副本D=4
ROLLBACK		
	commit	
		转储结束

转储过程

例:

转储开始

备份

数据库

A	B	C	D
1	2	3	4
1	7	6	4
1	7	6	4

T1	T2	备份
B:=7 A:=5 ROLLBACK	C:=6 commit	副本A=1
		副本B=2
		副本C=6
		副本D=4
		转储结束

转储过程

例:

转储开始

备份

数据库

A	B	C	D
1	2	3	4
1	7	6	4
1	2	6	4

T1	UNDO
T2	REDO

T1	T2	备份
B:=7	C:=6	副本A=1
		副本B=2
A:=5		副本C=6
		副本D=4
ROLLBACK	commit	转储结束

转储过程

(2) 海量转储与增量转储

- ❖ 海量转储: 每次转储全部数据库
- ❖ 增量转储: 只转储上次转储后更新过的数据
- ❖ 海量转储与增量转储比较
 - 从恢复角度看, 使用海量转储得到的后备副本进行恢复往往更方便
 - 如果数据库很大, 事务处理又十分频繁, 则增量转储方式更实用更有效



(3) 转储方法小结

❖ 转储方法分类

转储方式	转储状态	
	动态转储	静态转储
海量转储	动态海量转储	静态海量转储
增量转储	动态增量转储	静态增量转储

在数据转储效率、数据库运行效率、故障恢复效率三个方面各有利弊

DBA通常会根据数据库使用情况，确定一个适当的转储周期，并配合使用这类4类方法

10.4 恢复的实现技术

10.4.1 数据转储

10.4.2 登记日志文件



10.4.2 登记日志文件

1.日志文件的格式和内容

2.日志文件的作用

3.登记日志文件



1.日志文件的格式和内容

❖ 什么是日志文件

- 日志文件(log file)是用来记录事务对数据库的更新操作的文件

❖ 日志文件的格式

- 以记录为单位的日志文件
- 以数据块为单位的日志文件



以记录为单位的日志文件

❖ 以记录为单位的日志文件内容

- 各个事务的开始标记(**BEGIN TRANSACTION**)
- 各个事务的结束标记(**COMMIT**或**ROLLBACK**)
- 各个事务的所有更新操作

以上均作为日志文件中的一个日志记录 (log record)



以记录为单位的日志文件（续）

❖ 记录事务开始标记的日志记录

■ 事务标志 + **BEGIN TRANSACTION**

例：T1 **BEGIN TRANSACTION**



以记录为单位的日志文件（续）

❖ 记录事务结束标记的日志记录

- 事务标志 + COMMIT

- 事务标志 + ROLLBACK

例：T1 COMMIT

T2 ROLLBACK



以记录为单位的日志文件（续）

❖ 记录事务更新操作的日志记录

- 事务标识（标明是哪个事务）
- 操作类型（插入、删除或修改）
- 操作对象（记录ID、Block NO.）
- 更新前数据的旧值（对插入操作而言，此项为空值）
- 更新后数据的新值（对删除操作而言，此项为空值）

示意性例子: T1 U AA 18 20

 T1 I TU 1

 T1 D TV 20



以数据块为单位的日志文件

❖ 以数据块为单位的日志文件，每条日志记录的内容

- 事务标识
- 被更新的数据块



2.日志文件的作用

❖ 用途

- 进行事务故障恢复
- 进行系统故障恢复
- 协助后备副本进行介质故障恢复



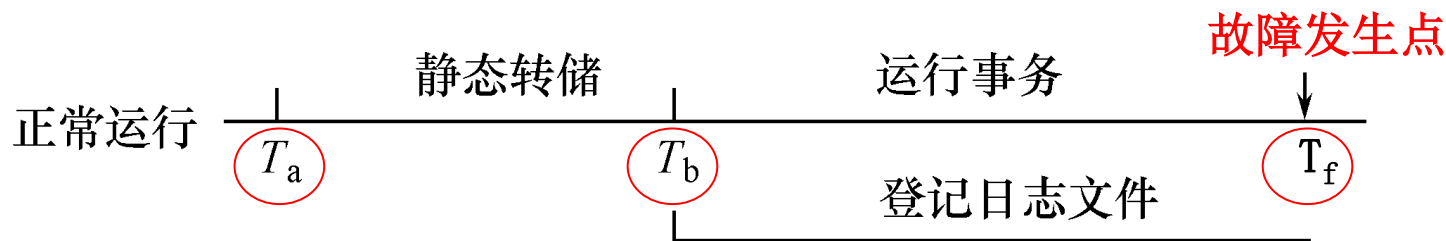
日志文件的作用（续）

❖ 具体作用

- 事务故障恢复和系统故障恢复必须用日志文件。
- 在动态转储方式中必须建立日志文件，后备副本和日志文件结合起来才能有效地恢复数据库。
- 在静态转储方式中，也可以建立日志文件。
 - 故障恢复时重新装入后备副本把数据库恢复到转储时刻的正确状态
 - 利用日志文件，重做已完成事务，撤销未完成的事务
 - 不必重新运行那些已完成的事务程序就可把数据库恢复到故障前某一时刻的正确状态



日志文件的作用（续）



利用静态转储副本与日志文件进行恢复



3.登记日志文件

❖ 为保证数据库是可恢复的，登记日志文件时必须遵循两条原则

- 登记的次序严格按并发事务执行的时间次序

- 必须先写日志文件，后写数据库

- 写日志文件操作：把表示这个修改的日志记录写到日志文件中

- 写数据库操作：把对数据的修改写到数据库中



登记日志文件（续）

❖ 为什么要先写日志文件

- 写数据库和写日志文件是两个不同的操作
- 在这两个操作之间可能发生故障
- 如果先写了数据库修改，而在日志文件中没有登记下这个修改，则以后就无法恢复这个修改了
- 如果先写日志，但没有修改数据库，按日志文件恢复时只不过是多执行一次不必要的**UNDO**操作，并不会影响数据库的正确性



小结

❖ 数据转储

- 静态转储 与 动态转储
- 海量转储 与 增量转储

❖ 登记日志文件

- 日志文件的格式和内容
- 日志文件的用途
- 登记日志文件时要遵守的原则



