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

故障的种类

- 事务内部的故障
- 系统故障
- 介质故障
- 计算机病毒

一、事务内部的故障

- 事务内部的故障
  - 有的是可以通过事务程序本身发现的（见下面转账事务的例子）
  - 有的是非预期的，不能由事务程序处理的。

事务内部的故障（续）

- 例如，银行转账事务，这个事务把一笔金额从一个账户甲转给另一个账户乙。

```
BEGIN TRANSACTION
 读账户甲的余额 BALANCE；
BALANCE=BALANCE-AMOUNT；  (AMOUNT 为转账金额 )
IF(BALANCE < 0 ) THEN
   {打印 '金额不足，不能转账 '；
   ROLLBACK；（撤销刚才的修改，恢复事务 )}
ELSE
   {读账户乙的余额 BALANCE1；
   BALANCE1=BALANCE1+AMOUNT；
     写回 BALANCE1；
COMMIT；  }
```

事务内部的故障（续）

- 这个例子所包括的两个更新操作要么全部完成要么全部不做。否则就会使数据库处于不一致状态，例如只把账户甲的余额减少了而没有把账户乙的余额增加。
- 在这段程序中若产生账户甲余额不足的情况，应用程序可以发现并让事务滚回，撤销已作的修改，恢复数据库到正确状态。

事务内部的故障（续）

- 事务内部更多的故障是非预期的，是不能由应用程序处理的。
  - 运算溢出
  - 并发事务发生死锁而被选中撤销该事务
  - 违反了某些完整性限制等

以后，事务故障仅指这类非预期的故障

事务内部的故障（续）

- 事务故障意味着
  - 事务没有达到预期的终点 (COMMIT 或者显式的 ROLLBACK)
  - 数据库可能处于不正确状态。
- 事务故障的恢复：撤消事务（UNDO）
  - 强行回滚（ROLLBACK）该事务
  - 撤销该事务已经作出的任何对数据库的修改，使得该事务象根本没有启动过一样

二、系统故障

系统故障

称为软故障，是指造成系统停止运转的任何事件，使得

系统要重新启动。

- 整个系统的正常运行突然被破坏
- 所有正在运行的事务都非正常终止
- 不破坏数据库
- 内存中数据库缓冲区的信息全部丢失

系统故障的常见原因

- 特定类型的硬件错误（如 CPU 故障）
- 操作系统故障
- DBMS 代码错误
- 系统断电

系统故障的恢复

- 发生系统故障时，一些尚未完成的事务的结果可能已送入物理数据库，造成数据库可能处于不正确状态。
  - 恢复策略：系统重新启动时，恢复程序要强行撤消（UNDO）所有未完成事务
- 发生系统故障时，有些已完成的事务可能有一部分甚至全部留在缓冲区，尚未写回到磁盘上的物理数据库中，系统故障使得这些事务对数据库的修改部分或全部丢失
  - 恢复策略：系统重新启动时，恢复程序需要重做（REDO）所有已提交的事务

三、介质故障

- 介质故障

  称为硬故障，指外存故障

  - 磁盘损坏
  - 磁头碰撞
  - 操作系统的某种潜在错误
  - 瞬时强磁场干扰

- 介质故障破坏数据库或部分数据库，并影响正在存取这部分数据的所有事务

- 介质故障比前两类故障的可能性小得多，但破坏性大得多

介质故障的恢复

- 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

四、计算机病毒

- 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，造成对计算机系统包括数据库的危害
- 计算机病毒种类
  - 小的病毒只有 20 条指令，不到 50B
  - 大的病毒像一个操作系统，由上万条指令组成

计算机病毒（续）

- 计算机病毒的危害
  - 有的病毒传播很快，一旦侵入系统就马上摧毁系统
  - 有的病毒有较长的潜伏期，机器在感染后数天或数月才开始发病
  - 有的病毒感染系统所有的程序和数据
  - 有的只对某些特定的程序和数据感兴趣
- 计算机病毒已成为计算机系统的主要威胁，自然也是数据库系统的主要威胁
- 数据库一旦被破坏仍要用恢复技术把数据库加以恢复

故障小结

- 各类故障，对数据库的影响有两种可能性
  - 一是数据库本身被破坏
  - 二是数据库没有被破坏，但数据可能不正确，这是由于事务的运行被非正常终止造成的。

恢复

- 恢复操作的基本原理：冗余
  - 利用存储在系统其它地方的冗余数据来重建数据库中已被破坏或不正确的那部分数据
- 恢复的实现技术：复杂
  - 一个大型数据库产品，恢复子系统的代码要占全部代码的 10% 以上

第十章 数据库恢复技术

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）
   - 登录日志文件（logging）
2. 如何利用这些冗余数据实施数据库恢复

10.4.1 数据转储

一、什么是数据转储

二、转储方法

一、什么是数据转储

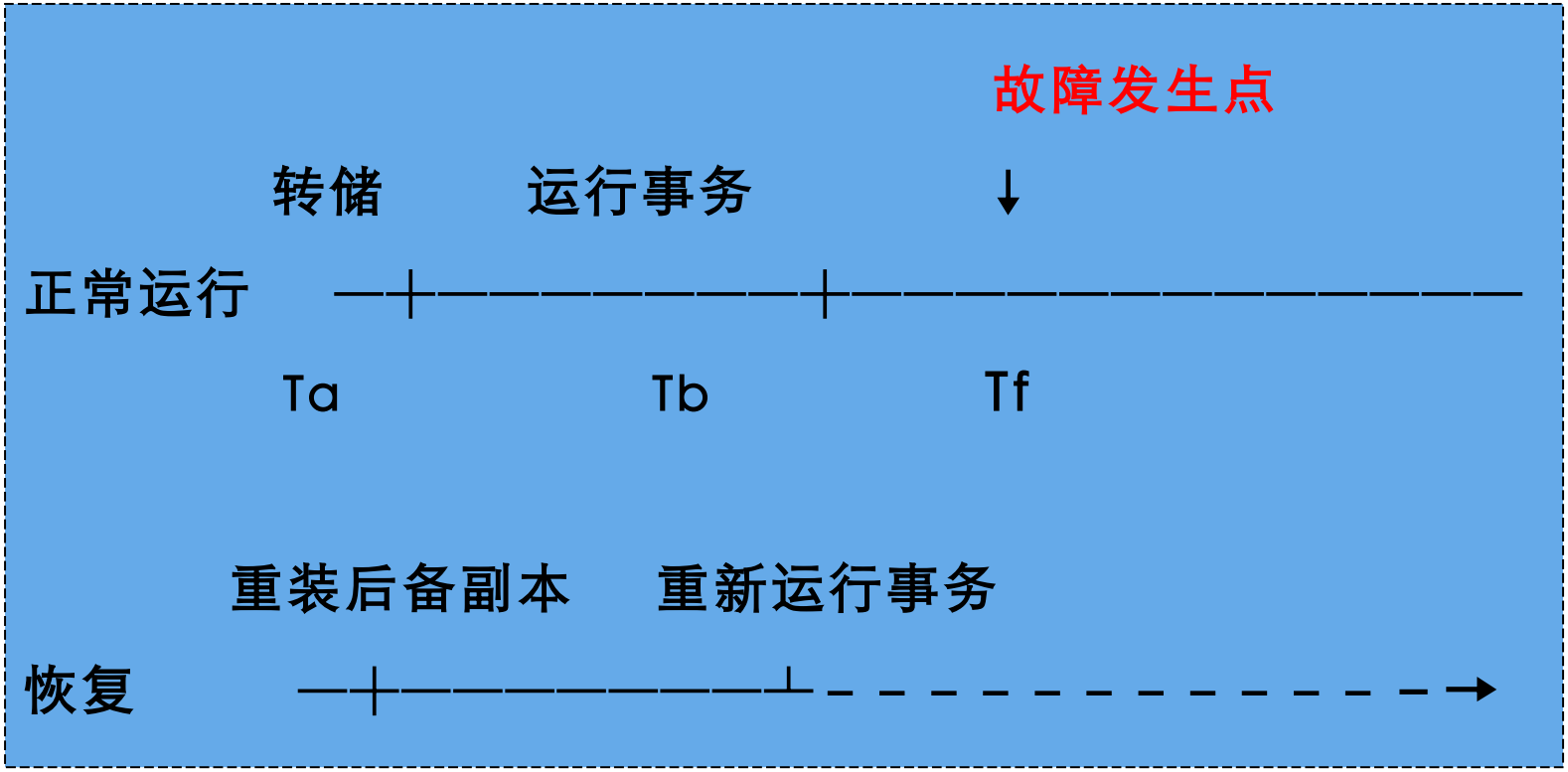
- 转储是指 DBA 将整个数据库复制到磁带或另一个磁盘上保存起来的过程
- 备用的数据文本称为后备副本或后援副本

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]



转储和恢复

数据转储（续）

上图中：

- 系统在 T_a 时刻停止运行事务，进行数据库转储
- 在 T_b 时刻转储完毕，得到 T_b 时刻的数据库一致性副本
- 系统运行到 T_f 时刻发生故障
- 为恢复数据库，首先由 DBA 重装数据库后备副本，将数据库恢复至 T_b 时刻的状态
- 重新运行自 T_b ～ T_f 时刻的所有更新事务，把数据库恢复到故障发生前的一致状态

二、转储方法

1．静态转储与动态转储

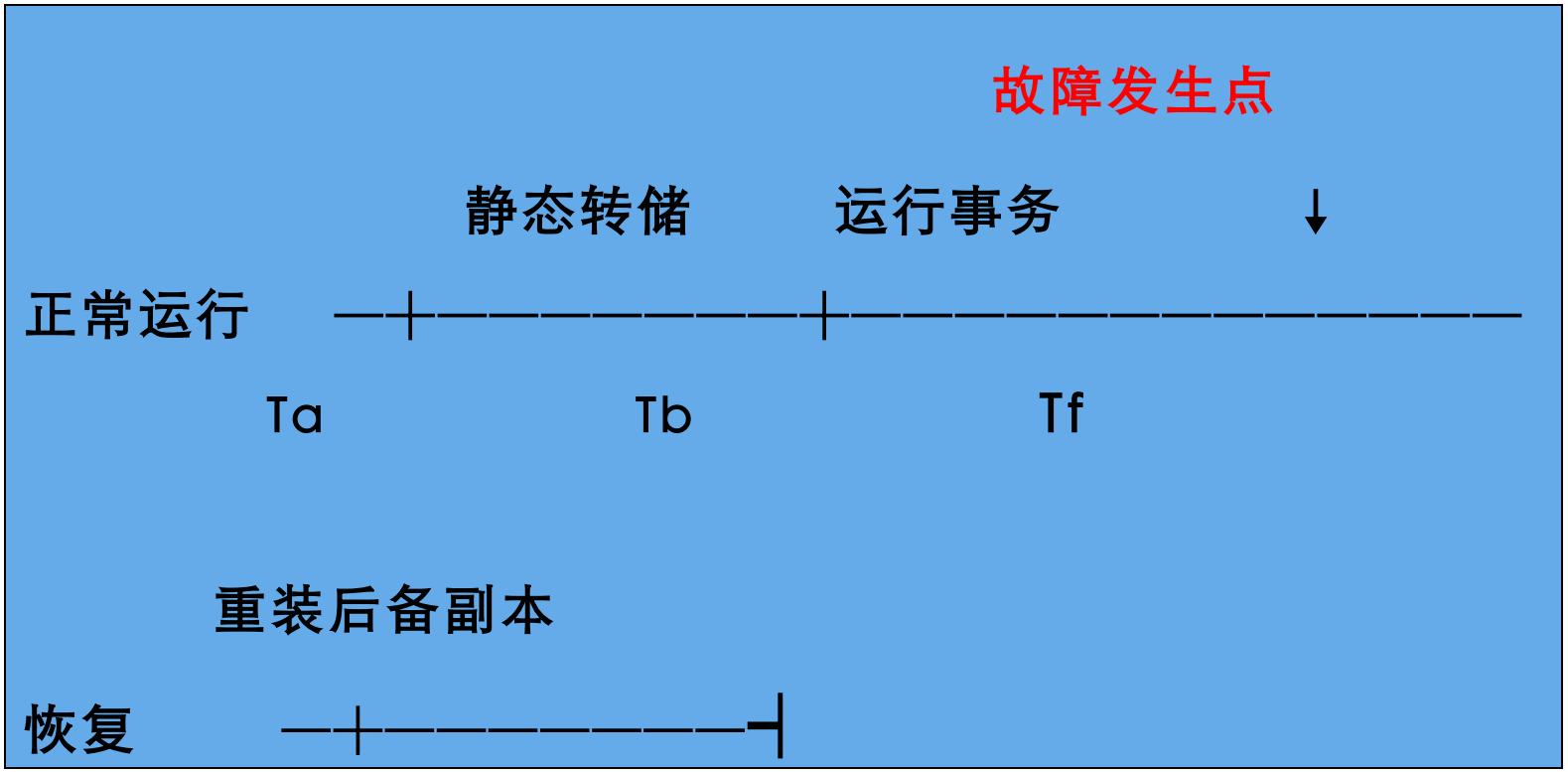
2．海量转储与增量转储

3．转储方法小结

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：实现简单
- 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

利用静态转储副本进行恢复



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
- 动态转储的缺点
  - 不能保证副本中的数据正确有效

例：设某张表由 A 、 B 、 C 和 D 构成

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

转储开始

正确数据

数据库备份

备份的是一个不一致的数据库状态

转储过程

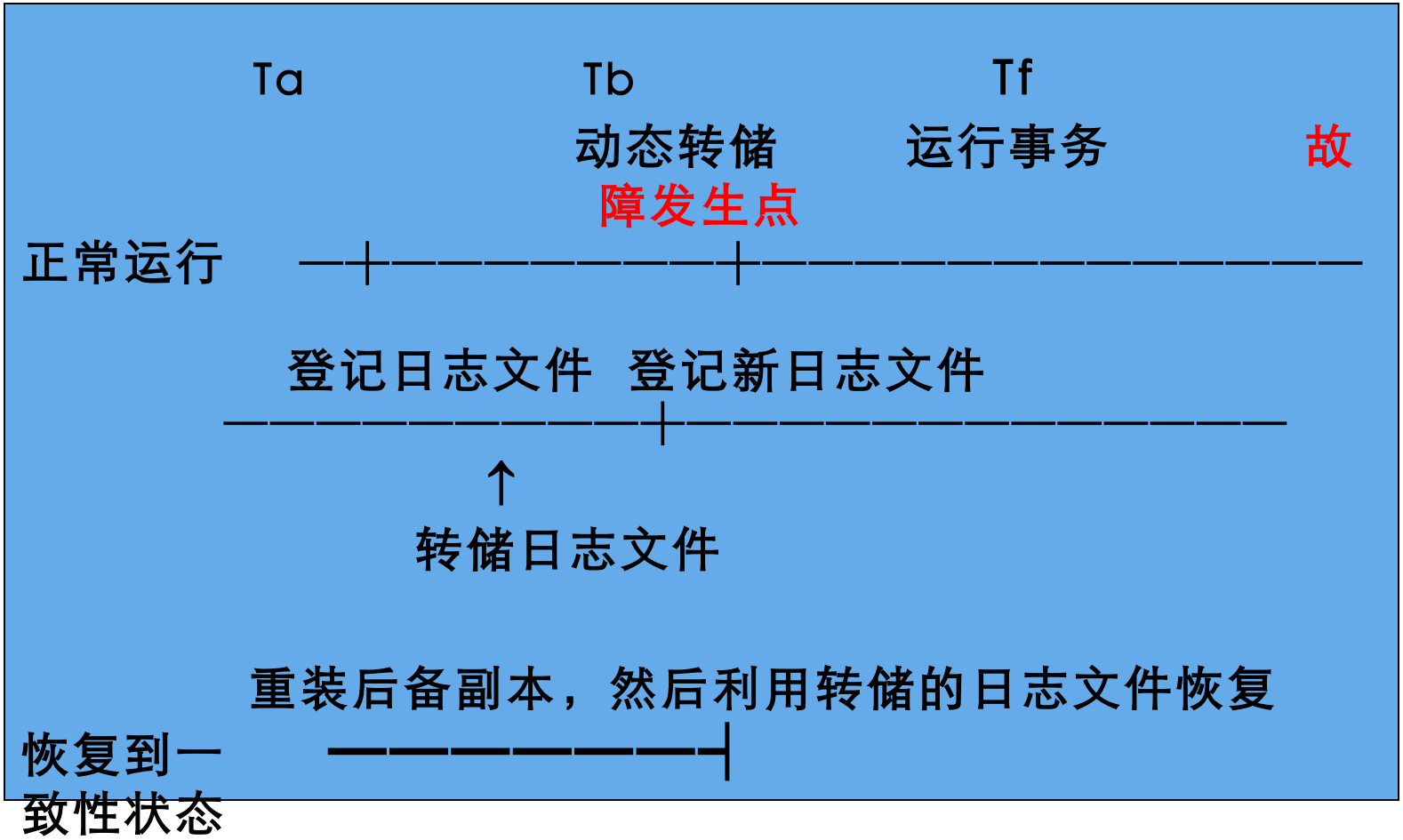
磁盘	备份
B:=7	副本A
C:=6	副本B
A:=5	副本C
	副本D

动态转储的问题

动态转储

- 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才能把数据库恢复到某一时刻的正确状态

利用动态转储副本进行恢复



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
  - 但如果数据库很大，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

		转储状态	
		动态转储	静态转储
转储方式	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

转储策略

- 应定期进行数据转储，制作后备副本。
- 但转储又是十分耗费时间和资源的，不能频繁进行。
- DBA 应该根据数据库使用情况确定适当的转储周期和转储方法。

  例：
  - 每天晚上进行动态增量转储
  - 每周进行一次动态海量转储
  - 每月进行一次静态海量转储

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

一、日志文件的格式和内容

二、日志文件的作用

三、登记日志文件

一、日志文件的格式和内容

什么是日志文件

日志文件 (log) 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

日志文件的格式和内容（续）

- 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)

日志文件的格式和内容（续）

- 以记录为单位的日志文件，每条日志记录的内容
  - 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）

日志文件的格式和内容（续）

- 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（标明是那个事务）
  - 被更新的数据块

二、日志文件的作用

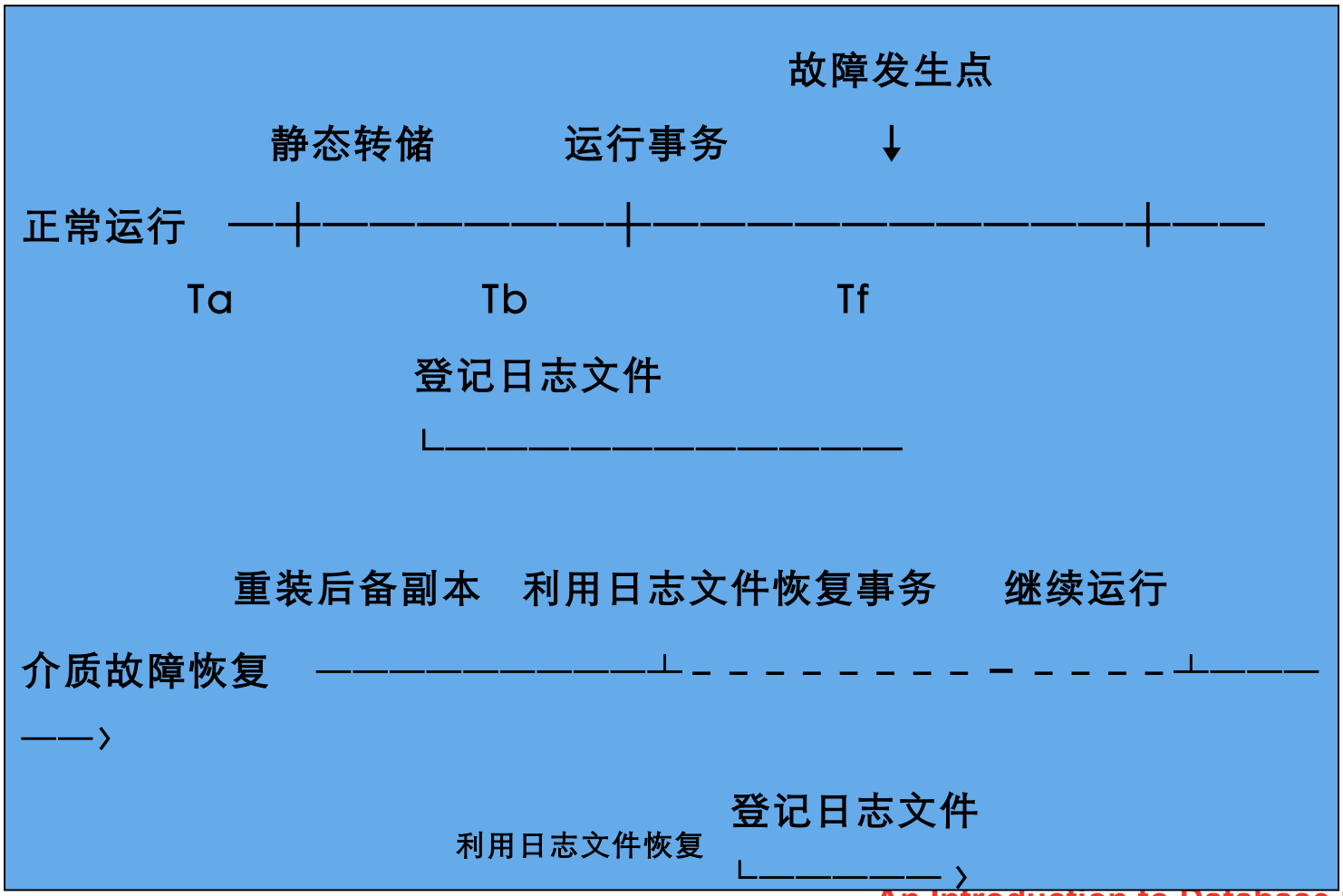
1．用途

- 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）

- 具体作用是：
  - 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。
    - 当数据库毁坏后可重新装入后援副本把数据库恢复到转储结束时刻的正确状态
    - 利用日志文件，把已完成的事务进行重做处理
    - 对故障发生时尚未完成的事务进行撤销处理
    - 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）


故障发生点
静态转储
运行事务
正常运行
Ta
Tb
Tf
登记日志文件
重装后备副本
利用日志文件恢复事务
继续运行
介质故障恢复
登记日志文件
利用日志文件恢复

三、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行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
    - 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