第十章 数据库恢复技术

事务是一系列的数据库操作，是数据库应用程序的基本逻辑单元

10.1 事务的基本概念

1、事务

事务是用户定义的一个数据库操作序列，事务是不可分割的工作单位

事务是恢复和并发控制的基本单位

（1） 事务的定义

显式：

BEGIN TRANSACTION	BEGIN TRANSACTION
SQL 语句 1	SQL 语句 1
SQL 语句 2	SQL 语句 2
◦◦◦◦◦	◦◦◦◦◦
COMMIT	ROLLBACK

COMMIT：事务正常结束，提交事务的所有操作，对数据库的更新重新写回到物理数据库

ROLLBACK：事务异常终止，事务回滚到开始时的状态

隐式定义:用户没有显式地定义事务，DBMS 按缺省自动划分事务

2、事务的 ACID 特性

（1）原子性 不可分割

（2）一致性

事务执行的结果必须是使数据库从一个一致性状态变到另一个一致性状态

一致性状态：数据库只包含成功事务提交的结果

不一致状态：事务执行发生故障，但是又一部分正确执行的已经写入数据库

一致性与原子性密切相关

（3）隔离性

一个事务的执行不能被其他事务干扰

并发执行的各个事务之间不能互相干扰

（4）持久性（也叫永久性）

一个事务一旦提交，它对数据库的改变是永久性的，接下来其他操作或故障不能

对其有影响

保证事务 ACID 特性是事务管理的任务

破坏事务 ACID 特性的因素：

（1）多个事务并行运行，不同事务的操作交叉执行 DBMS 要保证不影响隔离性

（2）事务在运行过程中被强行停止 要保证被停止事务不影响其他事务

10.2 数据库恢复概述

故障不可避免：计算机硬件、软件、操作员、恶意破坏

会导致事务非正常中断或是数据库被破坏

DBMS 必须有把数据库从错误状态恢复的某一已知的正确状态（或一致状态或完整状态）的功能。

10.3 故障的种类

1、事务内部的故障

有些事可以通过事务程序本身发现的（转账例子）

更多的是非预期的，不能由事务程序处理。如运算溢出、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而被终止。

事务故障即没有达到预期终点。需要强行回滚，即事务撤销

2、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

所有正在运行的事务都非正常终止，不破坏数据库，但缓冲区的信息全部丢失

常见原因：特定类型的硬件错误（如 CPU），操作系统故障，DBMS 代码错误，系统断电等

恢复策略：恢复程序让所有非正常终止的事务回滚，强行撤销所有未完成事务；还要重做已提交事务（有些提交了但是还未写入数据库）

3、介质故障

称为硬故障，指外存故障，如磁盘损坏、磁头碰撞、瞬时强磁场干扰等

将破坏数据库或部分数据库，并影响所有涉及的事务。这类可能性但小破坏性大

4、计算机病毒

一种计算机程序，小到不足 50B 大到由上万条指令组成

计算机病毒已成为数据库系统的主要威胁

故障总结：一、数据库本身被破坏

二、数据库没有但是数据不正确

恢复操作基本原理：冗余。利用冗余数据重建被破坏或不正确的数据

回复的实现技术：复杂。一个大型数据库产品，恢复代码要占到10%以上

10.4 恢复的实现技术

10.4.1 数据转储

1、什么是数据转储

数据转储是数据库恢复技术中采用的基本技术

备用的数据称为后备副本或后援副本

数据库遭到破坏后可将后备副本重新装入，但是只能恢复到转储时的状态。

2、转储方法

（1）静态转储与动态转储

静态转储是在无运行事务时进行的转储操作

静态转储得到的一定是一个一致性状态。但降低了数据库可用性

动态转储是指转储期间允许对数据库进行存取或修改，转储和事务可以并发操作。

使用动态转储必须使用**日志文件**把转储时各事务对数据库的修改活动记录下来

（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

比较：从恢复角度看，海量转储更方便

若数据库很大，事务处理又十分繁琐，增量转储更实用有效

（3）动态海量、静态海量、动态增量、静态增量

10.4.2 登记日志文件

1、日志文件的格式与内容

日志文件是用来记录事务对数据库的更新操作的文件

日志文件有两种：以记录为单位，以数据块为单位

（1）以记录为单位

日志文件中的内容：各个事务的开始，各个事务的结束，各个事务所有更新操作

以上剧作为日志文件中的一个日志记录

每个日志文件中的内容：事务标识（哪个事务）、操作类型（插入、修改、删除）、操作类型（记录内部标识）、更新前数据的旧值、更新后数据的新值

（2）以数据块为单位

每条日志记录的内容：事务标识、被更新的数据块

更新前后的数据块都放入日志文件中了，不需要操作类型对象等信息

2、日志文件的作用

（1）事务故障恢复和系统故障恢复必须用日志文件。

（2）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

（3）静态转储方式也可以建立日志文件，利用日志文件重做已提交的事务，对故障时未完成的事务做撤销处理

3、登记日志文件

两条原则：

（1） 登记次序严格按照并发事务执行的时间次序

（2） 必须先写日志文件，再写数据库

10.5 恢复策略

10.5.1 事务故障的恢复

事务故障是由系统自动完成，用户不需要干预

（1） 反向扫描日志文件，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，将更新前的值写入数据库。

（3） 继续反向扫描，查找该事务的其他操作

（4） 直达读到事务的开始标记，事务故障恢复完成

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有两个

（1） 未完成事务对数据库的更新已写入数据库

（2） 已完成事务放在缓冲区但是还没有写入数据库

恢复方法：Undo 故障发生时未完成的事务，Redo 已经完成的事务

（1） 正向扫描日志文件，列出两个队列

Redo 队列：找到故障发生时已经完成的事务（有 BEGIN 也有 COMMIT）

Undo 队列：故障发生时尚未完成的事务（有 BEGIN 没有 COMMIT）

（2） 对 UNDO 队列做 UNDO 处理，反向扫描，把更新前的值写入数据库

（3） 对 REDO 队列做 REDO 处理，正向扫描，把更新后的值写入数据库

10.5.3 介质故障的恢复

重装数据库->重做已完成的事务

（1）转入最新后备副本

若是静态的，直接恢复即可；若是动态的，用系统故障恢复方法恢复一致性状态

（2）装入有关的日志文件副本，重做已完成的事务

首先扫描日志文件，把提交的事务记入重做队列，然后正向扫描日志文件 REDO

介质故障的恢复需 DBM 的介入，重新装数据库和日志文件副本，执行恢复命令

10.6 具有检查点的恢复技术

1、问题的提出

搜索整个日志文件耗时，REDO 浪费大量时间

2、检查点技术

检查点记录的内容：建立检查点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

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

动态维护日志文件的方法：周期性检查检查点，保存数据库状态

（1） 将日志缓冲区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数据库中

（4） 把检查点记录的地址写入一个重新开始文件

检查点可以按照预定时间建立（如一个小时建立一个），也可以按照某种规则（如日志写满一半建立一个检查点）

3、利用检查点的恢复策略

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

T5：在检查点之后开始执行，在故障点时还未完成　　撤销

系统使用检查点进行恢复的步骤：

（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位置

（2） 由该检查点找到正在执行的事务清单

建立两个队列

REDO：需要执行 REDO

UNDO：需要执行 UNDO　把 ACTIVE-LIST 放入 UNDO 队列，REDO 为空

（3）从检查点扫描日志文件

如有开始的事务，则放入 UNDO 队列

如有提交的事务，则放入 REDO 队列

（4）分别执行

10.7 数据库镜像

1、问题：介质故障恢复比较耗时，DBM 必须周期转储数据库，麻烦

解决方案：数据库镜像（Mirror）

2、数据库镜像：DBMS 自动把整个数据库或是关键数据复制到另一个磁盘上。DBMS 自动保证镜像与主数据的一致性，主数据库更新，DBMS 自动复制过去

3、出现介质故障时，可以使用镜像磁盘，同时 DBMS 利用镜像数据恢复主数据

没有出现故障时，可用于并发操作

但是，频繁的复制数据会降低系统运行效率，实际中只选择复制关键数据和日志文件