



EagleBear2002 的博客

这里必须根绝一切犹豫，这里任何怯懦都无济于事

数据库系统概论-10-数据库恢复技术

📅 2022-06-16 | 📅 2025-11-26 | 📁 南京大学软件学院本科课程, 2022Spring-数据库系统概论 | 👁

121

📄 3.5k | ⌚ 3 分钟

1. 事务

事务 (Transaction, 缩写 txn) 是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, 是一个不可分割的工作单位。

事务是恢复和并发控制的基本单位。

事务的 ACID 特性:

- 原子性 (Atomicity): 事务是数据库的逻辑工作单位, 事务中包括的诸操作要么都做, 要么都不
- 一致性 (Consistency)
- 隔离性 (Isolation)
- 持续性 (Durability)

保证事务 ACID 特性是事务处理的任务。

破坏事务 ACID 特性的因素:

1. 多个事务并行运行时, 不同事务的操作交叉执行; 数据库管理系统必须保证多个事务的交叉运行不影响这些事务的隔离性
2. 事务在运行过程中被强行停止; 数据库管理系统必须保证被强行终止的事务对数据库和其他事务没有任何影响

1.1 事务定义

```
1 BEGIN TRANSACTION
2 COMMIT
3 ROLLBACK
```

COMMIT

1. 事务正常结束
2. 提交事务的所有操作（读+更新）
3. 事务中所有对数据库的更新写回到磁盘上的物理数据库中

ROLLBACK

1. 事务异常终止
2. 事务运行的过程中发生了故障，不能继续执行
3. 系统将事务中对数据库的所有已完成的操作全部撤销
4. 事务滚回到开始时的状态

1.2 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：数据库中只包含成功事务提交的结果。

不一致状态：数据库系统运行中发生故障，有些事务尚未完成就被迫中断；这些未完成事务对数据库所做的修改有一部分已写入物理数据库，这时数据库就处于一种不正确的状态。

1.3 隔离性

1. 一个事务的执行不能被其他事务干扰
2. 一个事务内部的操作及使用的数据对其他并发事务是隔离的
3. 并发执行的各个事务之间不能互相干扰

1.4 持续性也称永久性 (Permanence)

1. 一个事务一旦提交，它对数据库中数据的改变就应该是永久性的。
2. 接下来的其他操作或故障不应该对其执行结果有任何影响。

2. 故障和数据恢复

数据库管理系统必须具有把数据库从错误状态恢复到某一已知的正确状态（亦称为一致状态或完整状态）的功能，这就是数据库的恢复管理系统对故障的对策

故障的种类：

1. 事务内部的故障
2. 系统故障

3. 介质故障

4. 计算机病毒

2.1 事务故障

事务故障意味着：

1. 事务没有达到预期的终点（COMMIT 或者显式的 ROLLBACK）
2. 数据库可能处于不正确状态。

事务故障的恢复：事务撤消（UNDO）

1. 强行回滚（ROLLBACK）该事务
2. 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一样

2.2 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（特定类型的硬件错误（如 CPU 故障）、操作系统故障、数据库管理系统代码错误、系统断电），使得系统要重新启动。

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

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，恢复程序让所有**非正常终止的事务回滚**，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**。

3. 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、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。

4. 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。

5. 数据恢复

恢复操作的基本原理：冗余

利用存储在系统别处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。

6. 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。

静态转储：

1. 在系统中无运行事务时进行的转储操作
2. 转储开始时数据库处于一致性状态
3. 转储期间不允许对数据库的任何存取、修改活动
4. 得到的一定是一个数据一致性的副本

动态转储：

1. 转储操作与用户事务并发进行
2. 转储期间允许对数据库进行存取或修改

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。

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

7. 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。

用途：

1. 进行事务故障恢复
2. 进行系统故障恢复
3. 协助后备副本进行介质故障恢复

以记录为单位的日志文件内容：

1. 事务标识（标明是哪个事务）
2. 操作类型（插入、删除或修改）
3. 操作对象（记录内部标识）
4. 更新前数据的旧值（对插入操作而言，此项为空值）
5. 更新后数据的新值（对删除操作而言，此项为空值）

以数据块为单位的日志文件，每条日志记录的内容：

1. 事务标识
2. 被更新的数据块

7.1 日志文件的作用

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：

1. 当数据库毁坏后可重新装入后备副本把数据库恢复到转储结束时刻的正确状态
2. 利用日志文件，**把已完成的事务进行重做处理**
3. 对故障发生时**尚未完成的事务进行撤销处理**
4. 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

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：

1. 登记的次序严格按并发事务执行的时间次序
2. 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）

8. 恢复策略

8.1 事务故障的恢复

事务故障：事务在运行至正常终止点前非正常终止。

恢复方法：由恢复子系统利用日志文件撤销（UNDO）此事务已对数据库进行的修改

恢复步骤：

1. 反向扫描文件日志（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。

8.2 系统故障的恢复

系统故障造成数据库不一致状态的原因：

1. 未完成事务对数据库的更新可能已写入数据库
2. 已提交事务对数据库的更新可能还留在缓冲区没来得及写入数据库

恢复步骤：

1. 正向扫描日志文件（即从头扫描日志文件），将故障发生前已经提交的事务标记进入**重做队列**，将故障发生时尚未提交的事务标记进入**撤销队列**
2. 对撤销队列中事务进行撤销处理
3. 对重做队列中事务进行重做处理

8.3 介质故障的恢复

介质故障的恢复的工作：

1. 重装数据库（需要 DBA 介入）
2. 重做已完成的事务

恢复步骤：

1. 装入最新的后备数据库副本（离故障发生时刻最近的转储副本），使数据库恢复到最近一次转储时的一致性状态。
 - 对于静态转储的数据库副本，装入后数据库即处于一致性状态
 - 对于动态转储的数据库副本，还须同时装入转储时刻的日志文件副本，利用恢复系统故障的方法（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。
 - 首先扫描日志文件，找出故障发生时已提交的事务的标识，将其记入重做队列。
 - 然后正向扫描日志文件，对重做队列中的所有事务进行重做处理。即将日志记录中“更新后的值”写入数据库。

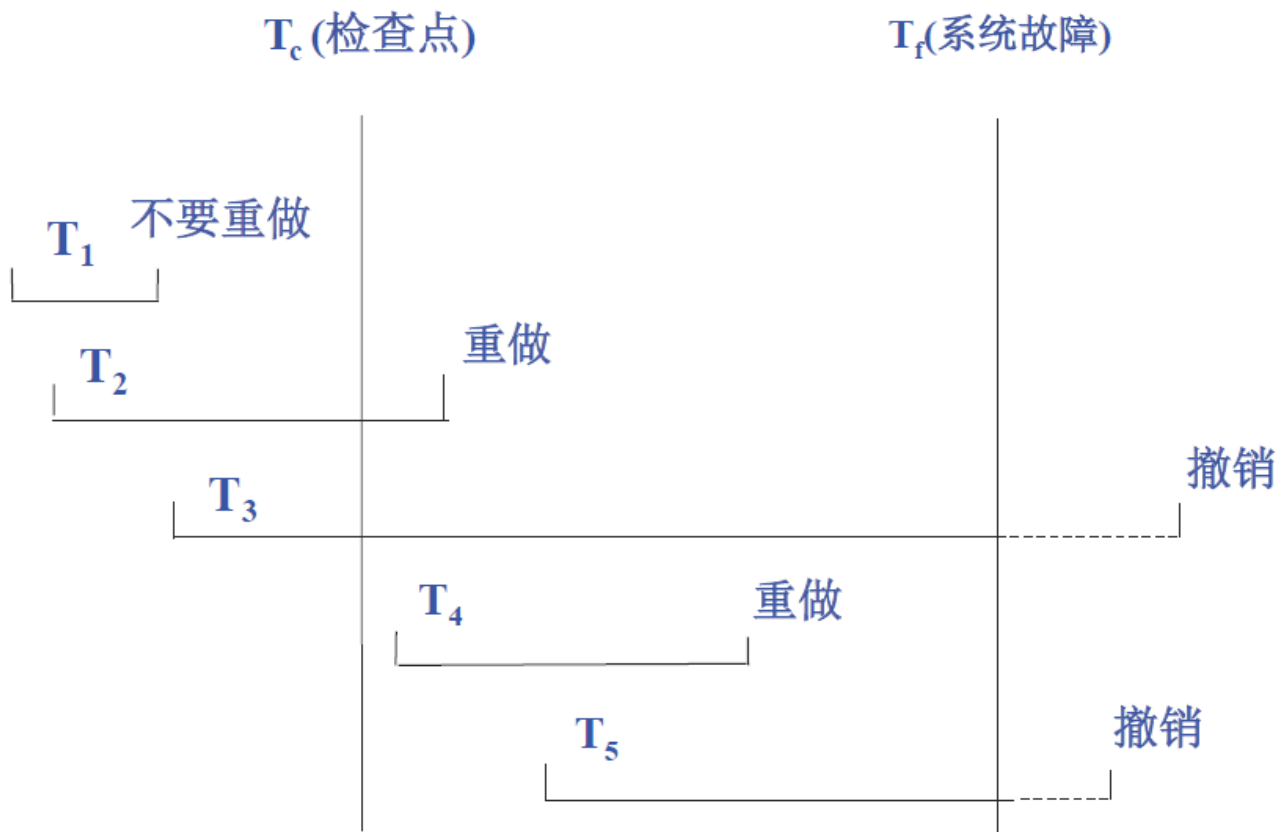
9. 具有检查点的恢复技术

9.1 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. 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. 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的数据库中
4. 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



9.2 利用检查点的恢复步骤

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。

1. 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，把 ACTIVE-LIST 暂时放入 UNDO-LIST 队列，REDO 队列暂为空。
2. 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；直到日志文件结束
3. 对 UNDO-LIST 中的每个事务执行 UNDO 操作
4. 对 REDO-LIST 中的每个事务执行 REDO 操作

打赏

原创

< 数据库系统概论-07-数据库设计

数据库系统概论-11-并发控制 >

© 2022 – 2025  EagleBear2002 |  2.7m |  40:13

由 [Hexo](#) & [NexT.Gemini](#) 强力驱动

 168530 |  444483