

《数据库系统》

作业 15

姓名: 刘韬

学院: 竺可桢学院

专业: 人工智能

邮箱: 3220103422@zju.edu.cn

日期: 2024-06-15

19.2

检查点是通过基于日志的恢复方案完成的，以减少崩溃后恢复所需的时间。如果没有检查点，则必须在崩溃后搜索整个日志，并且必须从日志中撤消/重做所有事务。如果执行了检查点，那么在恢复时可以忽略检查点之前的大多数日志记录。执行检查点的另一个原因是在稳定存储满时清除日志记录。

由于检查点在执行时会造成一些性能损失，如果快速恢复不是很重要，则应该减少检查点的使用频率。如果需要快速恢复，则应增加检查点频率。如果可用的稳定存储较少，频繁的检查点是不可避免的。检查点对从磁盘崩溃中恢复没有影响;归档转储相当于从磁盘崩溃中恢复的检查点。

19.10

交互式事务的恢复比批处理事务的恢复更加困难，主要原因在于交互式事务的即时性和参与者的直接参与。以自动柜员机（ATM）取款事务为例，这种交互式事务涉及到即时的用户交互和即时的物理现金交付，这些特点增加了恢复的复杂性：

1. 即时性和不可逆性：在 ATM 取款事务中，一旦现金被分发给用户，这个操作就是不可逆的。如果在事务过程中发生故障，如系统崩溃或通信中断，恢复过程需要确保用户账户的正确性（即确保账户金额正确调整），同时还要考虑到已经发放的现金无法回收的问题。
2. 外部参与者的直接参与：交互式事务通常涉及到外部参与者（如用户），这增加了事务的不确定性。例如，用户可能在事务过程中取消操作，或者在操作完成后立即使用取出的现金，这使得在发生故障时的恢复操作更加复杂。
3. 实时数据同步问题：在交互式事务中，如 ATM 取款，需要实时更新账户信息以反映取款操作。在发生系统故障时，必须确保所有数据都能正确同步，包括用户账户的减款和现金机内现金的减少。这需要复杂的恢复机制来处理可能的数据不一致问题。

相比之下，批处理事务通常在没有用户直接参与的情况下执行，可以在系统较不繁忙时进行，且通常涉及的是数据处理而非物理资源的交换，这使得其恢复过程相对简单。批处理事务可以设计为在发生故障时重新执行，而交互式事务如 ATM 取款则无法简单地重新执行，需要更细致和复杂的恢复机制。

19.21

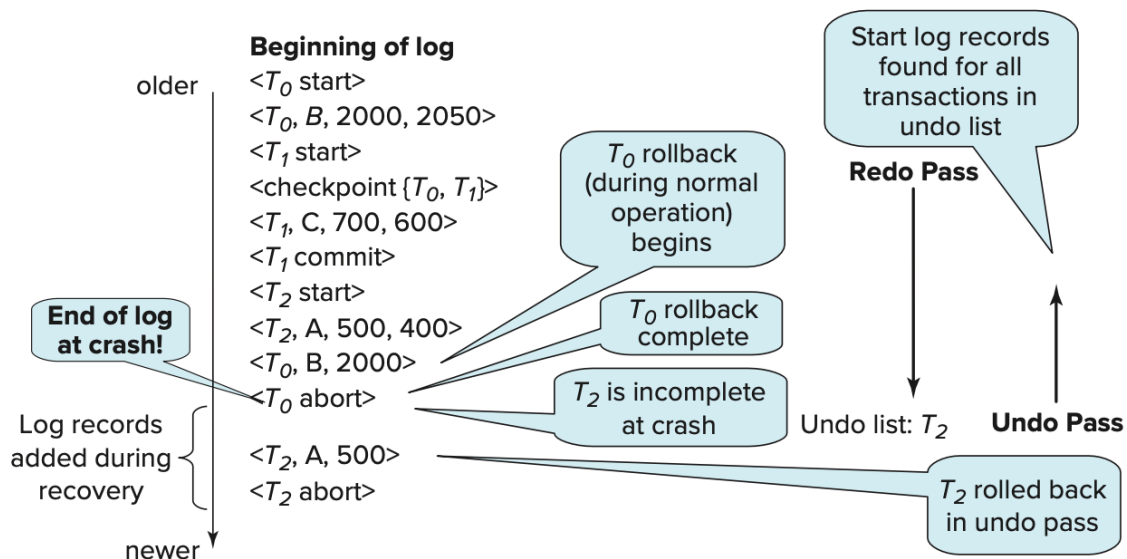


Figure 19.5 Example of logged actions and actions during recovery.

如果在 $\langle T_0, \text{abort} \rangle$ 之前发生 crash，那么在 undo list 中会有 T_0, T_2 ，因为 T_0 和 T_2 都是在 $\langle T_0, \text{abort} \rangle$ 之前提交的事务，需要 undo。那么在 recovery 过程中，会执行 undo 操作，将 T_0 和 T_2 的操作撤销，会先 $\langle T_0, \text{abort} \rangle$ ，然后其余的都不变。

19.25

在 ARIES (Algorithm for Recovery and Isolation Exploiting Semantics) 恢复算法中：

a. 如果在分析阶段的开始，一个页不在检查点脏页表中，我们通常不需要对它应用任何重做记录。原因是，检查点脏页表记录了在检查点时刻所有脏页（即已被修改但尚未写回磁盘的页）的信息。如果一个页在检查点时刻不在脏页表中，这意味着该页在检查点之前已经被成功写回到磁盘，因此在检查点之后的任何修改（即需要重做的操作）都不会影响该页。只有在检查点之后被修改并且在崩溃时仍然在内存中的页（即脏页）才需要应用重做记录。

b. RecLSN 是一个页头中的字段，代表了该页上最后一次被重做的日志记录的 LSN (Log Sequence Number)。RecLSN 用于减少不必要的重做，因为它允许系统确定一个给定的日志记录是否已经被应用到了该页上。在恢复过程中，如果一个重做日志记录的 LSN 小于或等于页的 RecLSN，这意味着该日志记录对应的修改已经被应用到了页上，因此可以跳过这个重做操作。这样，RecLSN 就可以有效地减少在恢复过程中不必要的重做操作，提高恢复效率。