

## 16.2

使用基于日志的恢复方案进行检查点，以减少恢复时间。如果没有检查点，则必须在崩溃后搜索整个日志，并从日志中撤消/重做所有事务。如果已执行检查点，则可以在恢复时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清除稳定存储中的日志记录。由于检查点在执行过程中会导致性能下降，因此如果快速恢复并不重要，则应降低其频率。如果我们需要快速恢复检查点频率应该增加。如果可用的稳定存储量较少，频繁的检查点是不可避免的。检查点对磁盘崩溃的恢复没有影响；归档转储相当于从磁盘崩溃中恢复的检查点。

## 16.10

交互式交易比批量交易更难以恢复，因为某些行为可能是不可撤销的。例如，输出（写入）声明可能会导致导弹，或导致银行机器向客户提供资金。处理此问题的最佳方法是尝试在事务结束时执行所有输出语句。这样，如果交易在中间中止，则不会造成任何伤害。理想情况下，输出操作应以原子方式完成；例如，ATM 机通常会计算笔记，并将所有笔记一起交付，而不是一次一个地发送笔记。如果无法以原子方式完成输出操作，则可以维护输出操作的物理日志，例如事件的磁盘日志，可以维护物理世界中发生的事件，以允许稍后手动执行恢复，例如通过将现金返还给客户帐号。

## 16.18

### Redo:

- Undo-List = T0, T1
- 从检查点条目开始并执行重做操作。
- C = 600 天。
- 由于存在提交记录，因此从撤消列表中删除 T1。
- 即在遇到<T2 start>记录时，T2 被添加到撤消列表中。
- A = 400。
- B = 2000。

### Undo:

- Undo-List = T0, T2
- 从末尾向后扫描日志。
- A = 500;输出仅重做记录<T2, A, 500>
- 输出<T2 abort>
- B = 2000;输出重做记录<T0, B, 2000>
- 输出<T0 abort>

在恢复过程结束时，系统的状态如下：A = 500 B = 2000 C = 600

恢复期间添加的日志记录为：<T2, A, 500> <T2 abort> <T0, B, 2000> <T0 abort>

观察到 B 由两个日志记录设置为 2000，一个在 T0 的正常回滚期间创建，另一个在恢复期间

创建，当中止 T0 完成时。尽管不是错误的，但是其次是冗余的。尽管描述了 ARAR 算法（在第 16.7 节中描述的等效优化）逻辑运算）可以帮助避免执行冗余操作，从而创建这样的冗余日志记录。

## 16.22

a. 如果在分析过程开始时页面不在检查点脏页表中，则不需要在检查点记录之前重做记录，以便在检查点之前将页面移到 todisk 并从 DirtyPageTable 中删除。但是，页面可能在检查点之后已更新，这意味着它将在分析过程结束时显示在 DirtyPageTable 中。对于出现在检查点 DirtyPageTable 中的页面，可能还需要应用检查点之前的重做记录。

b. RecLSN 是 Dirty PageTable 中的一个条目，当页面被添加到 DirtyPageTable 时，它会在日志末尾反映 LSN。在 ARIES 算法的重做过程中，如果遇到更新日志记录的 LSN 小于 DirtyPageTable 中页面的 RecLSN，则该记录不会重做但会被跳过。此外，重做传递从 RedoLSN 开始，这是检查点 DirtyPageTable 中条目中最早的 RecLSN，因为之前的日志记录肯定不需要重做。（如果检查点中没有 DirtyPage，则将 RedoLSN 设置为检查点日志记录的 LSN。）