## 16.2

检查点是通过基于日志的恢复方案来完成的，以减少崩溃后恢复所需的时间。
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。
由于检查点在使用时会造成一些性能损失，如果快速恢复不是关键，那么应该降低检查点的频率。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空间越来越少，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。

## 16.5

a. 不需要；如果事务已提交，则不再需要旧值，因为不需要撤消事务。如果在系统崩溃时事务是活动的，那么旧的值仍然安全地保存在稳定的存储中，因为它们还没有被修改；

b. 在重做阶段，不需要再维护撤销列表，因为稳定存储不会反映任何未提交事务导致的更新；

c. 读取数据项时首先向事务的本地内存发出读取请求。如果在那里找到，则返回。否则，该项将从数据库缓冲区加载到事务的本地内存中，然后返回；

d. 如果单个事务执行大量更新，则可能会耗尽用于存储数据项的本地副本的内存；

## 16.18

Redo phase:
a. Undo-List = T0, T1
b. 从检查点项开始，重做操作
c. C = 600
d. T1 从 Undo-List 里益处因为有一个提交记录
e. T2 被加入进 Undo-List 当遇到<T2 start>
f. A = 400
g. B = 2000

Undo phase:
a. Undo-List = T0, T2
b. 从结尾开始扫描日志
c. A = 500； 输出 redo-list 记录<T2, A, 500>
d. 输出<T2 abort>
e. B = 2000； 输出 redo-list 记录<T2, B, 2000>
f. 输出<T0 abort>

恢复后系统的状态：

A = 500
B = 2000
C = 600

恢复过程中增加的日志记录:
<T2, A, 500>
<T2 abort>
<T2, B, 2000>
<T0 abort>

## 16.20

Redo phase:
<T0, B, 2050>
<T0, C, 600>
<T1, C, 400>
<T0, C, 500>

Undo phase:
<T0, C, 400>
<T1, C, 600>
<T1, O2, operation-abort>
<T1 abort>
<T0, C, 700>
<T0, O1, operation-abort>
<T0, B, 2000>
<T0 abort>

最终 B=2000，C=700，与初始一致

## 16.22

a. 如果一个页面没有在检查站脏页表的开始分析,重做记录检查点记录前不需要适用于它，因为它意味着页面已经被刷新到磁盘,从之前的 DirtyPageTable 检查站被移除。但是，页面可能在检查点之后更新了，这意味着它将在分析结束时出现在脏页表中。对于出现在检查点脏页表中的页，可能还需要应用检查点之前的重做记录。

b. recLSN 是 DirtyPageTable 中的一个条目，它反映了当页面被添加到 DirtyPageTable 时日志末尾的 LSN。
在 ARIES 算法的重做过程中，如果遇到的更新日志记录的 LSN 小于 DirtyPageTable 中页面的 recsn，则该记录不重做，而是跳过。此外，重做通过从 RedoLSN 开始，它是检查点 DirtyPageTable 中最早的 recsn，因为早期的日志记录肯定不需要重做。(如果检查点中没有

脏页，则 RedoLSN 设置为检查点日志记录的 LSN。）