

Homework #15

19.2

1. System Performance When No Failure Occurs

- **Frequent Checkpoints:** 频繁执行检查点会带来巨大的开销，因为系统需要暂停或放慢常规操作来保存状态，降低系统的整体性能，尤其是在检查点进程需要大量资源的时候
- **Infrequent Checkpoints:** 降低检查点的频率可以减少这种开销，系统在正常运行时能够更高效地执行任务，减少保存状态信息所花费的时间

2. Time to Recover from a System Crash

- **Frequent Checkpoints:** 如果经常执行检查点，最近的检查点很可能接近故障点。系统崩溃后可以恢复到相对较近的状态，从而减少损失的工作量和恢复所需的时间
- **Infrequent Checkpoints:** 如果检查点少，最后保存的状态可能会过时，恢复需要重新处理大量数据和操作，增加恢复时间和数据丢失

3. Time to Recover from a Media (Disk) Failure

- **Frequent Checkpoints:** 与系统崩溃类似，频繁的检查点可确保磁盘上存储有系统状态的最新副本。这可以极大地促进磁盘故障后的恢复，因为系统可以使用这些最近的检查点快速恢复运行
- **Infrequent Checkpoints:** 如果检查点少，磁盘上的数据会更旧，恢复时间更长。而且需要处理和恢复的数据量更大，需要做更多时间才能恢复

19.10

- **实时处理:** 交互式交易，如ATM取款，实时进行。如果在交易过程中出现故障，很难确定交易的具体状态，确保用户操作正确回滚或完成
- **并发控制:** 多个用户同时操作系统，确保数据一致性和处理并发冲突增加了复杂性
- **即时效果:** 交互式交易往往有即时效果，如提款。如果在现金发放后但交易记录前出现故障，系统状态难以协调
- **不可预测:** 系统需处理不同用户行为，管理资源和保持稳定性更难

解决：加入 checkpoint、预写日志、两阶段提交

19.21

Redo phase:

(a) Undo-List = T0, T1

(b) Start from the checkpoint entry and perform the redo operation.

(c) Set C=600

(d) T1 is removed from the Undo-list as there is a commit record.

(e) T2 is added to the Undo list on encountering the < T2 start > record.

(f) Set A=400

(g) Set B=2000

Undo phase:

(a) Undo-List = T0, T2

(b) Scan the log backwards from the end.

(c) Set A = 500; write a log record < T2, A, 500 >

(d) Write a log record < T2, abort >; remove T2 from undo-list

(e) Set B = 2000; write a log record < T0, B, 2000 >

(f) Write a log record < T0, abort >; remove T0 from undo-list

(g) Undo-list is empty

19.25

a

脏页表跟踪检查点时已被修改但尚未写入磁盘的页面。如果某个页面不在 DPT 中，则该页面在上次检查点后未被修改。因此，与该页面对应的任何重做记录都会反映在磁盘上，可以避免冗余操作，优化恢复

b

Tracking Earliest Changes: RecLSN 保存在脏页表中，记录要对该页面进行重做操作的最早时间

Efficient Redo Scan: 在恢复的 redo 阶段，ARIES 会从 DPT 中找到的最小 RecLSN 开始扫描日志。只有相关的日志才会重做，减少了扫描日志记录的数量

Selective Redo Application: 对于重做的每条日志，ARIES 都会检查受影响页面的 RecLSN 是否小于 LSN。如果是，ARIES 会对该页面执行重做操作。反之 ARIES 会跳过对该页面的重做操作