

SimpleDB Lab3 (事务与锁机制)

实验目标:

实现基于锁的事务系统

实现事务的 ACID 属性

实现一个基于 NO STEAL/FORCE 的内存页替换策略

实现一个锁机制, 简单的死锁检测

关键类的实现:

1、PageLock 类

成员变量:

- private TransactionId [] Tid;//事务 ID
- private PageId [] Pid;//页面 ID
- private int [] ll;//锁级别, 1 为读锁, 2 为写锁。初始值为-1, 即无锁。
- 主要成员函数:
- public boolean readLockable(TransactionId tid, PageId pid), 判断页面时候可以加读锁;
- public boolean writeLockable(TransactionId tid, PageId pid), 判断页面是否可以加写锁;
- public void lockPage(TransactionId tid, PageId pid, Permissions perm), 对页面加锁;
- public void releaseLock(TransactionId tid, PageId pid), 释放 tid 事务占用的页面 pid;
- public void releaseLockTrans(TransactionId tid), 释放 tid 事务占用的所有页面;
- public boolean writeLocked(PageId pid), 判断 pid 页面是否加了写锁;
- public int lockLevel(TransactionId tid, PageId pid), 返回 tid 事务占用 pid 页时锁的等级。

2、BufferPool 中新增方法:

releasePage(TransactionId tid, PageId pid), 释放 tid 事务占用的 pid 页;

- transactionComplete(TransactionId tid), 将 tid 事务 dirty 过的页面写入磁盘;
- transactionComplete(TransactionId tid, boolean commit), 如果 commit, 将 tid 事务 dirty 过得页面写入磁盘, 如果 abort, 将 tid 事务 dirty 过得页面重新读入 BufferPool;
- holdsLock(TransactionId tid, PageId pid), 判断 tid 事务时候占据 pid 页;