

Lab5-Report

517030910412 陆晗

2020.6.5

Design decision

Lock strategy Locking granularity is based on page granularity. And I design a class pageLock for each page. In the class I record the shared locks and exclusive lock. And I make sure there will be at most one exclusive lock. And numbers of shared locks are no limit. And If transaction t is the only transaction holding a shared lock on an object o, t will upgrade its lock on o to an exclusive lock. And I implement some method to maintain the pagelock structure.

Deadlock detection policy I implement the denpendency graph to check if there is the deadlock. Each time, we cannot apply for the lock, I will modify the edges in the graph and do dfs(depth first search) to find whether there is a cycle. If there exsits, I will throw a TransactionAbortedException to abort it.

API No change for API.

Incomplete elements I think none. If there exits, thanks very much to inform me and I will fill the hole as soon as quickly.

Others I spent **five days** in finishing the code. And I spend some time to read the reference book because I begin the lab before the lecture. Apart from this, I learn something about java principle of multithreaded locking.

And Seniors about lab5 debugging skills gave me a lot of help. I cost some time to find the deadlock. And the multithreading test is hard to debug.