Lab2-Report

517030910412 陆晗

2020.4.20

Design decision

Eviction policy I use the LRU(least recent use) policy to decide which page to be evited. Because I think the FIFO(first in first out) method is not very reasonable and LRU is simple to implement. I use Map to record the pages in the Bufferpool with value. Each time the page is loaded to the buffer or be used, I will increase the value to capacity of the Bufferpool. Meanwhile, I will decrease other page value by 1 and make should they will not be below zero. If I need to evict one page, I will find the min value page.

Insertion I just follow the document and the content in the textbook. And I just need to implement the findLeafPage, splitLeafPage and splitInternalPage. FindLeafPage is simple and only need to compare the values to decide which son to search. Split is occured in the insertion when the leaf pages is full. And the implement is to decide how many to split and change the point in time. In SplitLeafPage, I should also change siblingId and copy the midkey up. In splitInternalPage, I should move the key instead of copy. That's may be a little different.

Deletion The B+ tree will find the tuples using findLeafpage. In order to make the tree balance, I implement the fuction to redistribute and merge. If there is enough to steal from partner, I will do redistribution. Otherwise, I will merge the node. And the process is the inverse of the split and is the same to the document.

Bonus Maybe no.

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 **three days** in finishing the code. It cost me nearly the whole day to understand the instruction and be familiar with the structure and the API. I also learned about whether Java functions are passed by value or by reference to make sure whether I should change the dirtypages in the implement function. As a result of my carelessness about code auto-complete, I spend much time in finding the error that I use hashmap.contain instand of containkey.So stupid!And the implementation of B+ tree is just a careful code implementation.