Saturday, May 4, 2019

# Lab2 Report

---

Jiangbei Li

## - Design Decision

- Page eviction
  - I use least recently used (LRU) policy for page eviction to discard the least recently used page first.
  - To realize LRU, I give each page in the buffer a counter with initial value 0. In each operation including delete, insert, getpage, if one page is not used, its counter will increase by 1. When evicting pages, I will find the page with the max counter and evict it.
  - LRU has a large (O(n)) time and space cost. But it has a high hit rate, since the most recently visited page is more likely to be visited in the future.
- Search in B+ Tree
  - From root, I find the entry with the least key which is larger than the field, then keep doing the same operation on its left child until I get the leave page of the field.
- Insert/delete Tuple in BufferPool
  - In class BufferPool, I called BTreeFile.insert/deleteTuple. Moreover I mark the page dirty and use getpage to set the counter to 0, then put the page back to buffer.
- splitLeafPage
  - First, I get an empty leaf page as the right sibling of the old page.
  - Then, use iterator to find the middle tuple. And transfer half tuples to the new page.
  - Next, set the right and left sibling of the new page and the old page.
  - Get an empty slot on the parent internal page and insert a new entry. I can conclude the new page and old page must have the same parent.
  - Finally, update the parent pointer pod the two pages, since the old page may change its parent.
- SplitInternalPage

    - First, get an empty internal page as the right sibling of the old page.
    - Then use iterator to find the middle entry. And transfer half entry to the new page.
    - Next, get an empty slot on the parent internal page and insert a new entry.
    - Finally, refresh the parent point of the children of the two pages.
  - Steal
    - Calculate the transfer number.
    - Using iterator and reverseiterator to transfer entry or tuple.
    - Update the parent entry and parent pointer.
  - Merge
    - Transfer all the entries and tuples from right to right.
    - Set the right page as empty.
    - Update parent points of the children and delete the parent entry of the leftPage.
- Changes
  - I made changes in the tuple class: overload the function tuple.equal()
  - If not, there will be an error in testSplitRootPage when set the buffer size to 1.
- Weakness
  - My eviction policy is time consuming and when call the same page too many times, the counter of other pages will be very large.
- Difficulty and time
  - I spent 2 days on the lab. A bug in the bufferpool/inserttuple disturbed me for a long time.