CS 2550 – Principles of Database Systems (Spring 2020)
Dept. of Computer Science, University of Pittsburgh

Assignment #4: Recovery, Physical Storage, Access Methods

Release: February 14, 2020 Due: 8PM, Thursday, February 27, 2020.

Goal

This is a *group* assignment. Please work on this assignment in the same group as that for HW2. Question 1 is to further practice recovery. The remaining questions are to practice different storage and indexing structures.

Description

Answer the following questions [for a total of 100 points]:

1. [10 points] Develop appropriate data structures and a recovery algorithm based on before images for rolling back aborted transactions whose execution is not strict and yet the lost update problem is eliminated. Assume no system or media failures.

2. [20 points (10 points each)] Suppose that we are using linear hashing on a file that contains records with the following search-key values:

   $$2, 4, 3, 7, 6, 14, 18, 9, 19, 11$$

   Suppose the hash function is $h_0(x) = x \ \ mod \ \ 4$ and buckets can hold three records, and no overflow buckets initially. After each split it allocates an overflow bucket to be used to avoid the next split.

   (a) Show the linear hash structure after each split.

   (b) Compare and contrast basic linear hashing without overflow buckets and this scheme, with respect to Big-O of search, insert without collision, insert with collision, delete without compaction, delete with compaction. Besides, please also compare the two methods with respect to split frequency (more or less), internal fragmentation (more or less), and number of pointers needed by each method.

3. [15 points] Show the extensible hashing structure that result from inserting the keys of the previous problem using the least significant bits. Suppose that buckets can hold three records.

4. [15 points] Construct a B$^+$-tree for the following input key values:

   $$6, 3, 13, 20, 14, 21, 2, 5, 23, 30, 2, 16, 22, 35$$

   Assume that the tree is initially empty and values are added in the order listed above. Construct B$^+$-tree for the cases where the degree is three. Show the B$^+$-tree in the following three stages: after inserting 2, 30 and after inserting all the keys.

5. [10 points] Suppose we have a database system that uses only the fixed length records to store tuples, for the traditional rotational disks, one idea to design the page organization scheme of the on-disk storage is to use slotted page approach. As Solid State Disk (SSD) drive technology matures, we would like to use SSD to store our tuples. If the slotted page schema is not appropriate to continue using with the SSDs, suggest one appropriate page organization scheme for SSDs.

6. [10 points] Consider a hard disk with the following characteristics: Disk rotation once every 12 ms. Seek time to move the head between adjacent tracks is 1.5 ms. Every track has 64 sectors (0 to 63) that are stored in linear order (same order the heads sees them). Assume that the head is positioned at the start of sector 4 on track 12 and the memory buffer is large enough to hold half of a track. How long will it take to transfer all of the sectors of track 12 to the corresponding sectors on track 14? Note that sectors do not have to be written in the same order as they are read.

7. [20 points (5 points each)] Modern disk drives store more tracks on the outer tracks than the inner tracks. Since the rotation speed is constant, the sequential data transfer rate is also higher on the outer tracks. The seek time and rotational delay are unchanged. Considering this information, explain good strategies for placing files with the following kinds of access patterns:

   (a) Frequent, random accesses to a small file (e.g., catalog relations).

   (b) Sequential scans of a large file (e.g., selection from a relation with no index).

   (c) Random accesses to a large file via an index (e.g., selection from a relation via the index).

   (d) Sequential scans of a small file.

Submission Guidelines

1. Only one member of the team should submit the assignment. The team member whose pitt_user_name appears earlier in the alphabetical order should submit the assignment.
2. You are required to submit **exactly one** PDF file, containing your answers to all questions under your **team_number** (e.g., team1.pdf). In addition to providing the answers, you are expected to: **include your team number, the names and pitt user names of the team members at the top of the PDF file**.
3. Note that your are required to use a graph editor (such as dia, MS-Word, MS-powerpoint MacDraw, idraw, xfig) to generate your diagrams. **Handwritten diagrams, even if they are scanned and submitted electronically, will not be accepted/graded.**
4. Submit your assignments through the Web-based submission interface (at the class web page `http://db.cs.pitt.edu/courses/cs2550/current.term/`). **It is your responsibility to make sure the assignment was properly submitted.**
5. Submit your assignment by the due date (**8PM, Thursday, February 27, 2020**). There is **no late submission**.

Academic Honesty

The work in this assignment is to be done *independently* by each group. Discussions with other students not in your group on the assignment should be limited to understanding the statement of the problem. Cheating in any way, including giving your work to someone else will result in an F for the course and a report to the appropriate University authority.