

CS587 – Database System Implementation – Winter 2018

Homework 1

Haomin He

1. Describe the function of the following parts of a database system (1-2 sentences for each):

Disk Space Manager: talks to disk controller and initiates data I/O operations, such as allocate, deallocate, read and write commands. Disk Space Manager tracks page locations and block usage internally; it links page number to physical location.

Buffer Manager: mediates between external storage and main memory. Buffer Manager brings pages into memory as they are needed for processing, and overwrites/replaces pages when they become obsolete and new pages are required.

File and Index Structures: show how data is stored and organized inside of database. On the conceptual level, a relational DBMS manages tables of rows and indexes. On the physical level, these data structures are implemented as files of records.

2. Describe the function of the following items in a buffer pool (1-2 sentences for each):

Pin Count: All database transactions are required to properly “bracket” page operations using pin() and unpin() calls. Proper bracketing enables the system keeps a count of active users accessing a page (pinCount).

Dirty Bit: A dirty bit is a bit indicates whether or not the corresponding page has been modified. If page is modified, dirty bit must be set to true. The bit indicates that its associated page has been modified and has not been saved to disk yet.

Referenced Bit (clock algorithm): gives pages a second chance to stay in the buffer pool. Initialize the referenced bit to true in the beginning. The system tries to find a replacement page in the buffer pool; if current page has referenced bit == false choose it. If current page has referenced bit == true, then change referenced bit to false and move on.

3. I have a small set of data to which you need fast read access. You have two disks.

Would you stripe the data across the disks or would you mirror the data on the disks?

Explain your answer. (1-2 sentences).

Answer: I would stripe the data across the disks because it would give me higher performance in reading access. Striping data across two disks would provide two times the bandwidth of a single drive.

4. If a database did not use pin/unpin calls in the buffer pool, what could go wrong? (2-3 sentences)

Answer: If a database did not use pin/unpin calls in the buffer pool, the system cannot keep a count of active users accessing a page. Without pinCount, we do not know which page is being used. We may replace a page that is being used by some users.

5. Describe the steps in the process of adding a variable length record to a page.

Answer: Inserting a variable length record needs to find an empty slot of size \geq the record size, such that the wasted space is minimal. In the slot directory, we set a pointer to the new record and set offset of record from start of data area in the corresponding slot. Compacting the remaining records to maintain a contiguous area of free space gets rid of holes produced by record deletion.

6. Give an example of a schema and a query for which a row store would be preferable to a column store. Explain your answer.

Answer:

Student(studentID, address, GPA)

Select * From Student;

In this schema, if we want to know all the information of each student. Row store is more preferable. Because in row store, data are stored in the disk tuple by tuple. We don't need to jump around to get all record information and put them together, like column store does.

7. (Grad Only) Relational databases are sometimes faster than no-sql systems and sometimes slower than no-sql systems. Describe in your own words, two key features of relational data management systems that enable them to (sometimes) have superior performance.

Answer: Relational data management systems have superior performance because they are well structured, each row contains a unique instance of data for the categories defined by the columns, support all kinds of join queries, and they are more flexible in terms of arbitrary joins and queries.

8. (Grad Only) For the FIFO buffer pool replacement algorithm, describe a situation where that algorithm works well and describe a situation where that algorithm does not work well.

Answer: FIFO is first-in, first-out, which organize buffers as a queue. It is good for one-time big file reading, because we can just dump/overwrite the old useless pages with new pages. FIFO is bad when we need to use old pages sometimes in the future. Because at that time, we may just have written out the page we need to the disk. Thus, we need to read in that page again, which causes additional disk I/Os.

References:

Lecture slides

https://en.wikipedia.org/wiki/Dirty_bit

https://web.stanford.edu/class/cs346/2015/notes/Lecture_One.pdf

<http://searchstorage.techtarget.com/definition/disk-striping>

<https://www.loginradius.com/engineering/relational-database-management-system-rdbms-vs-no-sql/>