Database System Implementation- CSE 507

Homework 3

Due date: March 22, 2016 11:59pm

Instructions:

- All submissions must be made through usebackpack site for this course (https://www.usebackpack.com/iiitd/w2016/cse507)
- Only one submission per team would be considered and graded. It would be assumed that all members of the team have participated equally and same score would be given to all members of the team.
- Your submission should have names of all the members of your team.
- Any assumptions made while solving the problem should be clearly stated in the solution.
- Question 2 and Question 3 are for teams of size 3. These questions will not be graded for teams for size 2 or less.
- Very Important: Your code should not have a directory structure. All files (code + dataset) should be present in just one folder. Note that this is absolutely crucial for grading this assignment.

Question 1 (100 points)

For this question, you would be implementing the DBMIN database buffer manager and compare its performance against LRU and MRU page replacement strategies in a simulated environment. You may use a programming language of your choice, but make sure that the TAs are able to grade it. Following are the implementation details of the simulation environment and the buffer manager algorithms.

Implementation of Buffers for queries:
Buffers are a fixed set of simulated memory locations. You may choose to implement it as a vector of a user defined type "mem location." During the simulation some of these locations would be holing a valid page which is being used. Some could be marked as "to be evicted" (in case of DBMIN). And others could just be "NULL." The number of buffers would be give as input.

Page Table:
Page table is a hash map, with key as the "Page Id" and value as the "memory location" in the buffer. The primary goal of the page table to point to the memory location where a given "page" is located (if at all) in the buffer. If a given "Page Id" is not found in the page table, then it would be considered as "page fault".

Simulated Dataset and queries:
For this assignment assume that your database has following three tables: (a) Employee, (b) Department and (c) Project. The tables given below show their statistics.

Table 1: Table Statistics

Table Name	# records	#disk blocks
Employee	10000	1000
Department	100	~~30~~ 25
Project	5000	500

Table 2: Index Statistics

Table Name	Column	Type	#levels
Employee	SSN	Secondary	3
Department	Dno	Secondary	2

Layout of tables and indexes:

(a) Assume that blocks corresponding to Employee, Department and Project tables are laid out sequentially one after another. This means first block of Employee table is located at the physical address "page 1," and first block of the department is located at the physical address "page 1001."

(b) All index files are located after the files containing the tables. Further assume that the index files laid out in the order given in Table 2. For same of simplicity assume that all the indexes have fan out of 30. You would have to compute the page ids of the index files.

Simulated Queries:

For this assignment, you are required to simulate the page reference behavior of following four queries.

1) Block nested loop join between Department and Employee with Department as the outer loop.
2) Block nested loop join between Department and Project with Department as the outer loop.
3) Single loop join (on Dnumber) between Department and Project with Project as the outer loop. Assume that Dno is the primary key of the Department table.
4) Single loop join (on SSN) between Department and Employee with Department as the outer loop. Assume that SSN is the primary key of the Employee table.

Note that you need not fully implement these query processing algorithms. You just need to implement "a skeleton structure" which would have the same page references as the full version. After implementing these "skeletons" run them on the previously mentioned simulated dataset and record the pages referenced in the order in which they would requested by the actual algorithm. These page references would be used to test the performance of DBMIN, LRU and MRU.

Implementation of DBMIN algorithm

(a) Your implemented should have appropriate data structures for maintaining global free list and locality set for the query. These datasets may stay outside the buffer space allocated for queries.

(b) For each of the above mentioned queries, appropriate maximum locality set size and page replacement algorithm should be initialized. For the purposes of this assignment, you may hard code these for each of the queries.

(c) Also you should have appropriate data structures to store the statistics required by the page replacement algorithm.

(d) In case the maximum locality set size > #buffers available, then set maximum locality set size to #buffers available.

Experimental Design:

Variable Parameter:

#buffers available for queries = **50, 100**, 200, 400, 600, 800, 1000, 1200.

Comparison Metric to be Recorded:

Measure the total number of page faults encountered by each of DBMIN, LRU and MRU algorithm for the previously mentioned queries on Employee, Department and Project table.

Deliverables for this question:

(a) Code for DBMIN, LRU and MRU

(b) Skeleton code for queries

(c) Dataset (page references for each of the queries)

(d) Final results of the experiments

(e) A brief explanation of trends

(f) Very Important: Your code should not have a directory structure. All files (code + dataset) should be present in just one folder. Note that this is absolutely crucial for grading this assignment.

Question 2 (15 points)

Consider schedules S1, S2, and S3 below. Determine whether each schedule is strict, cascade-less, recoverable, or non-recoverable. (Determine the strictest recoverability condition that each schedule satisfies.)

S1: r1(x); r2(z); r1(z); r3(x); r3(y); w1(x); c1; w3(y); r2(y); w2(z); w2(y);c2;c3

S2: r1(x); r2(z); r1(z); w1(x); r3(y); r3(x); w3(y); r2(y); w2(z); w2(y); c1; c3; c2;

S3: r1(x); w1(x); c1; r2(z); r3(y); w2(z); w3(y); c2; r5(z); r4(y); w5(z);c3; w4(y);a4;c5

Question 3 (15 points)

Which of the following schedules is (conflict) serializable? For each serializable schedule, determine the equivalent serial schedules.

(a) r1 (X); r2 (X); w1(X); r3(X); w2(X)

(b) r2 (X); r3 (X); w3(X); w1(X); w2(X)

(c) r3 (X); r1 (X); w3(X); r2(X); w1(X)

(d) r3 (X); r2 (X); r1(X); w3(X); w1(X)