November 22, 2018
Arnab Chakrabarti, Rihan Hai

Implementation of Databases (WS 18/19)

Exercise 4

Due until December 4, 2018, 10am.
Please submit your solution *in a single PDF file* before the deadline to the L^{2}P system!
Please submit solutions in groups of three students.

Exercise 4.1 (Cost Estimation) (9 pts)

Consider the following relational schema and SQL query. The schema captures information about employees, departments, and finances (organized on a per department basis).

$$Emp(\underline{eid}, did, sal, hobby)$$
$$Dept(\underline{did}, dname, floor, phone)$$
$$Finance(\underline{did}, budget, sales, expenses)$$

$$Emp[did] \subseteq Dept[did]$$
$$Finance[did] \subseteq Dept[did]$$

```
SELECT d.dname, f.budget
FROM Emp e, Dept d, Finance f
WHERE e.did=d.did AND d.did=f.did AND d.floor=1
AND e.sal>59000 AND e.hobby='yodeling'
```

Suppose that the following additional information is available: Unclustered B+-tree indexes exist on Emp.did, Emp.sal, Dept.floor, Dept.did, and Finance.did. Furthermore, assume that the costs for a tree traversal (from root to a leaf) are 3, additional costs for scanning leaves sequentially can be ignored. The system's statistics indicate that employee salaries range from 10,000 to 60,000, employees enjoy 200 different hobbies, and the company owns two floors in the building. All attribute values are uniformly distributed. There are a total of 50,000 employees and 5,000 departments (each with corresponding financial information, i.e., relation Finance has also 5,000 records) in the database. The DBMS used by the company has just one join method available: index nested loops.

1. For each of the query's base relations (Emp, Dept, and Finance) estimate the number of tuples that would be initially selected from that relation if all of the non-join predicates on that relation were applied to it before any join processing begins. **(4 pts)**

2. Given your answer to the preceding question, what would be the total cost for a query plan that first performs the selections on Emp, then joins Emp with Dept, and then the result is joined with Finance. Assume again that selections are done before join, projections are done as many and as early as possible, and that the indexes are used if applicable. You can also assume, that intermediate results do not have to be stored (either, they fit into memory or they are pipelined to the next operator). **(5 pts)**

Exercise 4.2 (Physical Database Design) (6 pts)

Given a relational table Student(sno,name,age,marstat) which is stored in an unsorted heap file with 1,000 pages (primary key is sno). Your system should be optimized for the following queries:

1. `Q1: SELECT * FROM Student WHERE sno = 534558`
2. `Q2: SELECT name,age FROM Student WHERE age > 19 AND age < 24`

How do you physically organize your database? Which indexes (clustered/unclustered) should be created to optimize the overall performance for both queries? (For the fan-out of tree index G we take 100.)

Exercise 4.3 (Cost Estimation) (9 points pts)

Consider a relation $R(a, b, c, d)$ containing 5,000,000 records, where each data page of the relation holds 10 records.

1. Suppose R is organized as a sorted file with indexes, and R is stored in $R.a$ order. There are three access paths:

   A1. Access the sorted file for R directly.

   A2. Use a clustered B+ tree index on attribute $R.a$.

   A3. Use a clustered hash index on attribute $R.a$.

   For each of the following selection queries, state which of the three access paths is most likely to be the cheapest and explain why.

   (a) $\sigma_{a=50,000}(R)$ **(1 pts)**

   (b) $\sigma_{a\neq 50,000}(R)$ **(1 pts)**

   (c) $\sigma_{a>50,000 \wedge a<50,010}(R)$ **(1 pts)**

2. Assume that all four attributes of R are string fields of the same length. There are 1000 buffer pages. A projection query $\pi_{a,b}(R)$ should be executed. 20 records of the resulting relation can be stored in one page. Consider the optimized version of the sorting-based projection algorithm: The initial sorting pass reads the input relation and creates sorted runs of tuples containing only attributes a and b. Subsequent merging passes eliminate duplicates while merging the initial runs to obtain a single sorted result (as opposed to doing a separate pass to eliminate duplicates from a sorted result containing duplicates).

(a) How many sorted runs are produced in the first pass? What is the average length of these runs? (Assume that memory is utilized well and any available optimization to increase run size is used.) What is the I/O cost of this pass (**including** writing of the intermediate result)? **(2 pts)**

(b) How many additional merge passes are required to compute the final result of the projection query? What is the I/O cost of these additional passes? Writing of final result is **excluded**. **(2 pts)**

3. Suppose the following query with a self-join on R is executed: $\pi_{a,b}(R \bowtie_{a=a} R)$. Note that a is not a key attribute. Which join method (block-nested loop join, hash join, or sort-merge join) using any of the access paths from above is most likely to perform best. Why? What are the estimated costs? **(2 pts)**

Exercise 4.4 (Answer questions briefly) (6 pts)

1. Reason below formulas **in detail** **(2 pts)**

   (a) Why is the cost for an equality selection using a clustered tree index $D * (\log_G 0.15B)$?

   (b) Why is the cost for an insert operation using an unclustered hash index $4D$?

   Base your explanation on the below assumptions from empirical studies:

   - In a sorted file, pages are stored sequentially, retrieving a desired page directly only needs one disk I/O.
   - In a clustered file, pages are usually 67% full, and the number of physical data pages is 1.5B.
   - We omit the time for processing a record in memory (since it is usually negligible compared with the time for reading or writing disk pages)
   - D is the cost for one I/O operation.

2. Explain the CAP Theorem. **(2 pts)**

3. What is the default join method in Spark? Explain how it works. **(2 pts)**