Cost-based plan selection

Exercises

2013-2014

Exercise 1. Relation $R(a, b, c, d)$ has a clustered index on attribute a and unclustered indexes on the other attributes. The following statistics are given: $B(R) = 1000$, $T(R) = 5000$, $V(R, a) = 20$, $V(R, b) = 1000$, $V(R, c) = 5000$ and $V(R, d) = 500$. What is the best physical query plan (index scan or table scan, possibly followed by a filter) and cost for each of the following selections?

- $\sigma_{a=1 \text{ AND } b=2 \text{ AND } d=3}(R)$
- $\sigma_{a=1 \text{ AND } b=2 \text{ AND } c\geq 3}(R)$
- $\sigma_{a=1 \text{ AND } b\leq 2 \text{ AND } c\geq 3}(R)$

Exercise 2. Consider the following relations storing information on Employees (E), Departments (D) and Finances (F):

- E(eid: `int`, did: `int`, sal: `int`, hobby: `char`(20))
- D(did: `int`, dname: `char`(20), floor:`int`, phone: `char`(10))
- F(did: `int`, budget: `int`, sales: `int`, expenses: `int`)

An E-record is 35 bytes long, a D-record is 40 bytes long, and an F-record is 15 bytes long. Blocks are 2048 bytes in size, and we have 10 main memory buffers available. There are unclustered BTree indexes on `E.did` and `D.floor`. There are clustered BTree indexes on `E.sal`, `D.did` and `F.did`. The statistics show that the employee salaries are within the range $[10000, 60000]$ (uniformly distributed), that all employees together have 200 distinct hobbies, and that the company possesses 2 floors in the building. There are 50000 employees and 5000 departments (each with associated financial information) in the database.

The query compiler has already constructed the following logical query plan:

$$\pi_{\texttt{D.dname,F.budget}}(\sigma_{\texttt{E.hobby='yodeling' AND E.sal } geq 59000}(E) \bowtie \sigma_{\texttt{D.floor}=1}(D) \bowtie F)$$

Construct a sufficiently optimal physical query plan. Use disk I/Os (instead of the number of tuples as done in the book) as your optimization metric. Motivate your answer and describe any assumptions that you make. It suffices to consider only locally-optimal decisions (in other words: you may use the greedy algorithm and your solution need not be globally optimal.)

Exercise 3. Consider the following relations storing information on Employees (E), Departments (D) and Projects (P):

- E(eid: `int`, did: `int`, age: `int`, sal: `int`)
- D(did: `int`, pid: `int`, budget: `int`, status: `char`(10))
- P(pid: `int`, code: `int`, report: `string`)

An E-record is 20 bytes long, a D-record is 40 bytes long, and every P-record is 2000 bytes long. There are 20000 records in E, 5000 records in D and 1000 records in P. The projects are uniformly distributed over the departments. An employee can belong to only one department. A project can belong to only one department. Blocks are 4000 bytes in size and there are 12 main memory buffers available. The following questions are all based on this information. You may assume that all values are uniformly distributed. Mention any additional assumptions that you make. Use disk I/Os as your cost metric. Explain your answers.

1. Consider the query $\sigma_{\texttt{age}\geq 30}(E)$ when we have an unclustered index on `age`. Let N denote the number of tuples that satisfy the query. From which value of N onwards is a table scan followed by a filter cheaper than an index scan?

2. What join algorithm is to be preferred if we want to join E and D? Would your answer change if instead we have 51 buffers available? And if we would have a clustered BTree index on E.did?

3. Construct the optimal physical query plan for

$$\pi_{\texttt{D.did,COUNT}(*)}\gamma_{\texttt{D.did,COUNT}(*)}(\pi_{D.did,D.pid}(D) \bowtie \pi_{P.pid}(P))$$

   (Note that the inital projections can be done "on the fly" when computing the join.) Indicate when you materialize subresults. What is the total estimated cost? Compare your solution to the situation in which we have 51 buffers available instead. Also compare your solution to the situation where we have clustered BTree indexes on `E.did`, `E.pid` and `P.pid`.

4. Construct the optimal physical query plan for:

$$\pi_{\texttt{D.did,COUNT}(*)}\gamma_{\texttt{D.did,COUNT}(*)}(\pi_{D.did,D.pid}(\sigma_{\texttt{D.budget}\geq 99000}(D)) \bowtie \pi_{P.pid}(P))$$

   Assume that the department budgets are uniformly distributed in the range $[0, 100000]$. Indicate when you materialize subresults. What is the total estimated cost? What would the optimal plan be when we have a clustered BTree index on the composite search key `(D.did, D.budget, D.pid)`, and an unclustered index on `P.pid`?

5. Construct a sufficiently optimal physical query plan for:

$$\pi_{\texttt{E.eid,D.did,P.pid}}\sigma_{\texttt{E.sal}=50000}(E) \bowtie \sigma_{\texttt{D.budget}\geq 20000}(D) \bowtie P$$

   Assume that employee salaries are uniformly distributed over the range $[10009, 110008]$ and that project budgets are uniformly distributed over $[10000, 30000]$. There are clustered indexes available on `E.sal`, `D.did` and `P.pid`. It suffices to make only locally-optimal decisions (in other words: you may use the greedy algorithm and your solution need not be globally optimal.)