Database Tuning, ITU, Spring 2010

Rasmus Pagh

March 1, 2010

1 Representation of relations

In this problem we consider a relation `R(a,b)`, where `a` and `b` are integers (of type `INT`). We let $B > 1$ denote the number of integers that fits in a disk block. Suppose that R consists of N tuples, $\{(a_1, b_1), \ldots, (a_N, b_N)\}$, sorted such that $a_1 < a_2 < a_3 < \cdots < a_N$. There are two natural ways of representing the relation on disk, ordered according to `a`:

Horizontal: $a_1, b_1, a_2, b_2, \ldots, a_N, b_N$ (this is the standard order).

Vertical: $a_1, a_2, \ldots, a_N, b_1, b_2, \ldots, b_N$.

Some DBMSs ("column stores") allow the use of vertical order. We assume that there are no updates to the data, and it is thus stored as a sequential file. The size N of the relation is known.

a) How many I/Os are needed to read the K smallest values of `a`, i.e., $a_1, \ldots, a_K$, in each of the two representations? State your answers in terms of K and B.

b) How many I/Os are needed to read the K smallest values of `b` in each of the two representations? State your answers in terms of N, K, and B.

c) Assume that there in no index on `R`. How many I/Os are needed to find the tuple with a particular value of `a` in each of the two representations? State the worst case number of I/Os for the best algorithms you can think of.

We now consider a third alternative representation, the *multi-sorted* representation. Assume that $\sqrt{N}$ is an integer. The idea is to change the horizontal representation by partitioning it into $\sqrt{N}$ intervals of $\sqrt{N}$ tuples, and sorting each interval according to the value of `b`. An example instance with $N = 9$ is the following (we mark tuples by parentheses and intervals by square brackets for readability):

$$[(3,2),(2,3),(5,5)],[(5,4),(13,9),(11,10)],[(23,1),(19,6),(17,14)]$$

d) Show that in the multi-sorted representation, it is possible to search for a particular value of `a`, as well as a particular value of `b`, in at most $\sqrt{N} \log_2 N$ I/Os (without any index).

A similar effect can be obtained in DBMSs that allow indexes to be partitioned.

2 Indexing in DB2

Find answers to the following question in the DB2 indexing documentation (link on course home page):

1. What is the standard index type created by `CREATE INDEX`?
2. Does DB2 automatically create an index on any primary key? If not, what is the syntax for creating a primary index?
3. Does DB2 allow partitioned indexes?