DATABASE MANAGEMENT SYSTEMS

SPRING 2023

CS – 632 PROJECT(Theory Part)

Professor: Mr. Buchi Okoli Okoli

Project Members

Kaleeswaran Sivasankaran, James Helloween, Ravi Shankar Dhwarakesh

1)

We have a file with a million pages (N = 1,000,000 pages), and we want to sort it using external merge sort. Assume the simplest algorithm, that is, no double buffering, no blocked I/O, and quicksort for in-memory sorting. Let B denote the number of buffers.

How many passes are needed to sort the file with N = 1,000,000 pages with 6 buffers?

The number of pages per pass is equal to B-1, which in this case is 6-1 = 5.

Number of pages = ceil (log 5 (ceil(1000000/5))) + 1;

= ceil (log 5 (ceil (166667)) + 1 = 9;

2)

To find all keys between 9 and 19 in the given B+ tree, **5 pointers need** to be chased. Starting with the root node, all keys in the root's left subtree (10) are smaller than 9. As a result, the first node in the right subtree is 20, which has no keys between 9 and 19. However, because the root's second child (30) is greater than 19, there is no need to traverse its subtree. The only node that contains keys between 9 and 19 is the left child of the root (10).

From the root to the left child -> right sibling (20) –> left child (11) -> right sibling (12) -> right sibling (13)

3) **23**

24 – 11000

Inserting 24 will not cause any split as it will insert in 00 where the space is left out to add the element.

23 - 10111

As the last 2 is 11 in 23 and since there is no space left out in the last part, split will occur when inserting 23.

4) 15 nodes

A sparse B+ tree of order d = 2 containing the keys 1 through 20 inclusive would have 15 nodes. The tree would have 10 leaf nodes at level 1, 2 internal nodes at level 2 and 1 root node at level 3

Nodes (1, 2), (3, 4), (5, 6), (7, 8), (9, 10), (11, 12), (13, 14), (15, 16), (17, 18), (19)

Nodes (5, 9, 13), (17)

Root node (9, 17)

5)

The second logical plan (II) is the more efficient than the first one (I). In plan II, the selection operation $\sigma_{c=3}$ is applied to relation S before the join operation $\bowtie_{b=b}$ with relation R. This reduces the number of tuples that need to be joined, potentially reducing the cost of the join operation. In contrast, in plan I, the selection operation is applied after the join operation, which means that all tuples from both relations R and S are joined before the selection is applied.

6)

False

In the vectorized processing model, each operator that receives input from multiple children does not require multi-threaded execution to generate the Next() output tuples from each child. Instead, in the iterator model, the Next() function couples dataflow with control flow. When Next() returns a batch of tuples in the dataflow graph, control is returned too. As a result, the entire query plan can be executed with only one thread

7)

Hash join algorithm can be optimized by minimizing the hash table size by choosing the smaller of the two input tables as the build input, which is used to create the hash table. This can reduce the memory footprint of the hash table and improve performance. Another way is to use a good hash function that distributes the keys evenly among the hash buckets, reducing the hash collisions and improving performance. Furthermore, memory management can increase performance by minimizing the requirement for disk I/O through techniques such as using a multi-pass method that spills intermediate results to disk when memory use surpasses a particular threshold.

8)

1. Scanning the Applicants table costs 100 I/Os and produces an output of 100 tuples.
2. Scanning the Schools table costs 10 I/Os and produces an output of 9 tuples.
3. An in-memory sort-merge join is performed on the outputs from steps 1 and 2, producing an output of 9 tuples.
4. An index-nested loop join is performed on the output from step 3, costing 9 I/Os.
5. Steps 5 and 6 are done on-the-fly and do not incur any additional I/Os.

The total cost is 100 + 10 + 9 = 119 I/Os.

9)

a) One approach to improve the performance of the join operation is to use a rehashing technique. This involves creating hash tables for the inner and outer relations and then rehashing the large buckets into an embedded hash table using a second hash function.

b) B = 75 pages in the buffer

Table R spans M = 2,400 pages with 80 tuples per page

Table S spans N = 1,200 pages with 100 tuples per page

M + (M/B-2) * N = 2400 + (2400/75-2) * 1200 = 42852;

10.

The number of leaf nodes in a full binary tree with 2n internal nodes is n+1

11.

a.) Insert 12, Insert 13