COMP9120 Relational Database Systems

Tutorial Week 12: Query Processing

Introduction

Suppose we have a schema Rel1(A, B, C) and Rel2(C, D). Each A field occupies 4 bytes, each B field occupies 12 bytes, each C field occupies 8 bytes, each D field occupies 8 bytes. Rel1 contains 100,000 records, and Rel2 contains 50,000 records. There are 100 different values for A represented in the database, 1000 different values for B, 50,000 different values for C, and 10,000 different values for D. Rel1 is stored with a primary (sparse, clustered) B+ tree index on the pair of attributes (A,B); assume this index has 2 levels, including the root page and excluding the leaf (record) pages. Rel2 is stored with a primary (sparse, clustered) B+ tree index on C; assume this index has 2 levels, including the root page and excluding the leaf (record) pages.

General features: assume that each page is 4K bytes, of which 250 bytes are taken for header and array of record pointers. Assume that no record is ever split across several pages. Assume that index entries in any index use the format of (search key, rowid), where rowid uses 4 bytes.

Exercise 1. Block-Nested Loops Join

Consider the following query:

```
SELECT Rel1.A, Rel1.B, Rel1.C
FROM Rel1, Rel2
WHERE Rel1.C = Rel2.C AND Rel2.D = 16;
```

and consider the query plan which calculates this as follows:

Form the equi-join of Rel1 and Rel2 by using a block-nested loops join with Rel1 as the outer relation, and then filter each tuple of the join to see if the value of D is 16; if so, output the values of A, B and C from that tuple of the join.

How many page I/Os are needed to compute this plan (assume that we have only the minimal space, say 2 pages worth, for buffering in memory)?

Exercise 2. Index-Nested Loops Join

Reconsider the query in the above Exercise. But, now consider a different query plan, where the join is processed as an index-nested loops join (using the primary index on Rel2.C), and then each tuple of the join is filtered to check the value of D. How does this affect the cost?

1. Calculate the # of pages (we use 75% here)

Rel①. Rel(A, B, C) = 4 + 12 + 8 = 24 — 100,000 records

Rel②. Rel(C, D) = 8 + 8 = 16 — 50,000 records

- Each page is 4K, so 4 × 1024 = 4096 bytes
- 250 bytes for header
- So each page can have 4096 − 250 = 3846 bytes

Rel① $\lfloor 3846/24 \rfloor^{\times 0.75} = \lfloor 160 \times 0.75 \rfloor = 120$

Rel② $\lfloor 3846/16 \rfloor^{\times 0.75} = {}^{0.75\times} 240.375 = 180$

反正在后面用了所以不在乎前面式里面乘0.75都行

① → $\lceil \frac{100000}{120} \rceil = 834$

② → $\lceil \frac{50000}{180} \rceil = 278$

Ex 1:

use Rel(c,d) as inner relation

$834 + \boxed{834 \times 278} =$

Ex2:

- Assume index was not given 看过类似
- Firstly let's calculate the # of layers, and index

For Rel①

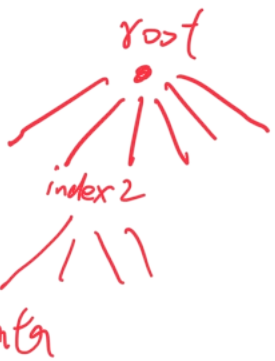
$$\frac{3846}{4+12+4} = \lfloor 192.3 \rfloor = 192 \times \underbrace{0.75}_{\text{assumption}} = 144$$

A: 4, B: 12, rowID: 4

$$\frac{834}{144} = \lceil 5.79 \rceil = 6$$

lowest level of Tree, we have 6 pages

Tree 1



6<144 所以只有2层

Rel②

$$\left\lfloor \frac{3846}{8+4} \times 0.75 \right\rfloor = 240$$

8: only c as PK

$$\left\lceil \frac{278}{240} \right\rceil = \lceil 1.15 \rceil = 2$$

→ 2 < 240 → 2 layers in total

↑ 如果是1则总共只有1 layer 即 root 就行

· Now use B+ trees structure, we have

First scan whole rel1: 834, set unique C then

Second Using C: go through tree1

· go through 3 pages (3 nodes) by using idx

· 100,000 × 3 = 300000

Total 834 + 300000 =