Database Systems

Lab 6

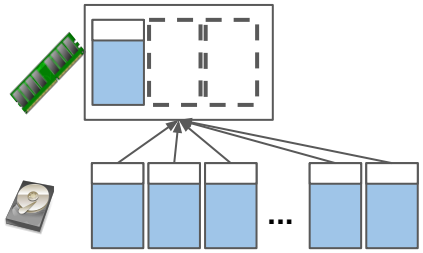
Today

- Recap
- Sort & Join Cost Estimation

Select

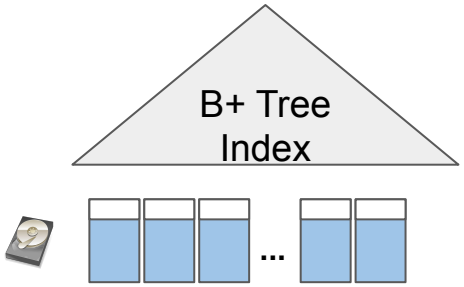
Trade-offs: *selectivity* is important

DBMS will choose for you



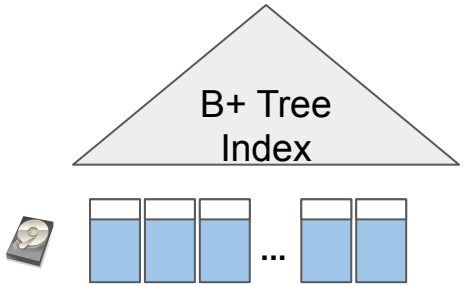
B(R)

Bad, but not always the worst



log(B(R)) + α(C,R).|R|

(unclustered)



log(B(R)) + α(C,R).B(R)

(clustered)

External Sort

Pass 0: output **B-page** files
Pass 1: output **B(B-1)-page** files
Pass 2: output **$B(B-1)^2$-page** files
…
Pass $\lceil \log_{B-1} \lceil (N/B) \rceil \rceil$: output **N-page** files

passes: $1 + \lceil \log_{B-1} \lceil (N/B) \rceil \rceil$ passes
One pass: 2.N

Total cost: **$2.N.(1+ \lceil \log_{B-1} \lceil (N/B) \rceil \rceil)$**

Join Recap

Example
R = 100,000 S = 40,000
B(R) = 1,000 B(S) = 500
M = 100

Algorithm	**Cost**	**Time**
Nested Loop	B(S) + |S|.B(R)	40,000,500 I/Os = **4000s**
Block Nested Loop	B(S) + B(R).⌈B(S)/(M-2)⌉	6,500 I/Os = **0.65s**
Sort Merge	B(S) + B(R) + sort(R) + sort(S)	5,849 I/Os = **0.58s**
Grace Hash	3(B(R) + B(S))	4,500 I/Os = **0.45s**

4 orders of magnitudes

DBMS will choose for you

Exercise 1

Given 2 relations R and S.

- R has 100 pages, 100 records per page.
- S has 50 pages, 50 records per pages. Suppose we have 20 buffer frames.

[Q1] What is the I/O cost of joining R and S with nested loop join, using S as outer relation.

[Q2] Same as Q1, but with block nested loop join

Notation

|R|, |S|: # tuples in R, S

B(R), B(S): # pages for R,S

M: buffer size

Exercise 2

Suppose you need to sort 108-page file using 4 buffer frames, using external sort.

[Q1] How many passes are needed?

[Q2] What is the total IO cost?

[Q3] What is the smallest number of buffer frames needed to sort the file in 2 passes?

Exercise 3

Suppose you have 10,000 buffer frames to sort a relation R, and R has 186 pages.

How many I/O does it cost?

Exercise 4

Index Join

```
for each tuple in R
      search index of S, output matches
```

Suppose you have 2 relations:

- R: 20,000 tuples; 25 tuples fit in a block
- S: 45,000 tuples; 30 tuples fit in a block
- Buffer size M=30
- R is sorted

[Q1]: What is the cost of sort-merge join the two relations?

[Q2]: Suppose that S has an ***unclustered*** B+-tree index on the join attribute, and the entire index is in memory.
What's the cost of the Index Join algorithm above. Assume each tuple in R matches with **k** tuples in S