

Class 1S: Joins I

Summary

#pages M

$\binom{N}{P_S}$

#rows per page P_R

S (sid, sname, rating, gpe)

$N = 500 \quad P_S = 80$

Selection

(a) i) cheapest access path

ii) retrieve tuples

iii) apply remaining selection conditions

R (sid, bid, day, rname)

$M = 1000 \quad P_R = 100$

(b)

i) get rows from all matching indexes

ii) intersection ride

iii) retrieve tuples & apply remaining sel. and.

unsorted M I/Os

sorted $\log_2 M + f \cdot M$

clustered $\log_F M + f \cdot M$

unclustered $\log_F M + f \cdot M \cdot P_R$

Projection

Sort + discard unwanted fields & duplicates
hash + CC

cost: $M + 2 \cdot T_{\leftarrow \text{Pages}}$ after removing unwanted fields

Joins :

Nested-Loop Joins

Sort-Merge Joins

today

Hash Joins

Remaining op (joins + agg)

any interesting query contains a join

SELECT * FROM R, S WHERE R.sid = S.sid

R \bowtie S discuss as # I/Os discard output

~~$\sigma_{R.sid=S.sid}(R \times S)$~~

Simple Nested-Loop Join

R \bowtie S
 $i=j$

$\forall r \in R$ \leftarrow outer

$\forall s \in S$ \leftarrow inner

if $r_i = s_j$ then add $\langle r, s \rangle$ to the result

Cost

$$(M \cdot P_R) \cdot N + M = (1000 \cdot 100) \cdot 500 + 1000 = \boxed{50,001,000} \text{ # I/Os}$$

rows of R \downarrow 1 I/O \rightarrow 2ms

$\boxed{28h}$

$$R \quad M=1000 \rightarrow 4MB$$

$$S \quad N=500 \rightarrow 2MB$$

Swap R with S

$$(N \cdot P_S) \cdot M + N = \boxed{40,000,500} \text{ I/Os}$$

Page-oriented Nested-Loop Join

\forall page br in R

\forall page bs in S

\forall tuple r in br

\forall tuple s in bs

if $r_i = s_j$ then add $\langle r, s \rangle$ to the result

Cost

$$M \cdot N + M = 1000 \cdot 500 + 1000 = \boxed{501,000} \rightarrow \boxed{17 \text{ min}}$$

Smaller outer?

$$N \cdot M + N = 500 \cdot 1000 + 500 = \boxed{500,500}$$

Index Nested Loop Join

tuple r in R

probe index to fetch s such that $s_i = r_j$

add $\langle r, s \rangle$ to result

Cost

$M + M \cdot PR \cdot \text{cost of finding matching tuples through the index}$

- Hash index 1.2 I/Os
- B*-Tree 2-4 I/Os

clustered $\rightarrow 1$ I/O per page of matching tuples
unclustered $\rightarrow 1$ I/O per matching tuple

Example 1: hash idx on sid of S

Scan R: (M)

each tuple in R

fetch data entry (1.2)

goto file (1)

$$M + M \cdot PR \cdot (1.2 + 1) \rightarrow 1000 \cdot 100 \cdot (2.2) = \boxed{221,000} \rightarrow \boxed{7 \text{ min}}$$

Example 2: hash idx on sid of R

$$\frac{M \cdot P_R}{N \cdot P_S} = \frac{1000}{400} = 2.5 \text{ res Sailor}$$

Scan S (n)

if s probe hash idx (1,2)

find matching tuples 2.5

$$N = N \cdot P_S (1.2 + 2.5) = 500 + 500 \cdot 80 \cdot (3.7) = \boxed{148,500} \rightarrow \boxed{5 \text{ min}}$$

Block Nested Loop Joins

→ 1 page for streaming the inner S

1 page for output

K pages for holding blocks (of K) of outer R

if block of K pages of R

if page bs in S

if tuple r in K pages of R

if tuple s in bs

if $r_i = s_j$ add $\langle r, s \rangle$ to the result

COST

Scan outer R; M I/Os

Scan inner for each block of R

$$M + \frac{M}{K} \cdot N \rightarrow 1000 + \frac{500 \cdot 1000}{K} \quad K = 100 \text{ pages}$$

$$\rightarrow \boxed{6000 \text{ I/Os}} \quad \boxed{12 \text{ sec}}$$

$$N + \frac{N}{K} \cdot M \rightarrow 500 + \frac{500 \cdot 1000}{K} \rightarrow \boxed{5500} \rightarrow \boxed{11 \text{ sec}}$$

Sort-Merge Join

→ both sorted on the join attribute

- useful: ① both or one relations sorted on join attr.
- ② output should be sorted on join attr.

→ many duplicates may lead to backtracking

Cost Sort R + Sort S + M + N

worst case? M · N if all is equal

cost $(M+N) \cdot 2 \cdot \text{passes} + M + N$

2 passes?

$$\lceil \frac{N}{B} \rceil = B-1 \approx \frac{N}{B} = B-1 \Rightarrow B^2 - B - N = 0$$

$$B \approx \sqrt{N+1} + 1 = 33$$

$$\text{cost} = (M+N) \cdot 5 = 1500 \cdot 5 = \boxed{7500} \text{ I/Os} \rightarrow \boxed{15 \text{ sec}}$$

BNLJ w/ 33 buffers

$$M + \frac{M \cdot N}{K} = 1000 + \frac{500 \cdot 1000}{33} \approx \boxed{1000 + 15151}$$

$$N + \frac{M \cdot N}{K} = 500 + \frac{500 \cdot 1000}{33} \approx \boxed{500 + 15151}$$

if $K=100$ SMJ cannot do better than $\boxed{7500}$

BNLJ will do as low as $\boxed{5500}$

Refined Sort-Merge Join

assume $B > \sqrt{M}$ and $B > \sqrt{N}$

after pass 0

$$R \rightarrow \frac{M}{B} \text{ runs} \quad B > \sqrt{M} \Rightarrow \frac{1}{B} < \frac{1}{\sqrt{M}} \Rightarrow \frac{M}{B} < \sqrt{M} < B$$

$$S \rightarrow \frac{N}{B} \text{ runs} \quad B > \sqrt{N} \Rightarrow \frac{N}{B} < \sqrt{N} < B$$

after pass 0 either R, S # runs $\leq B$

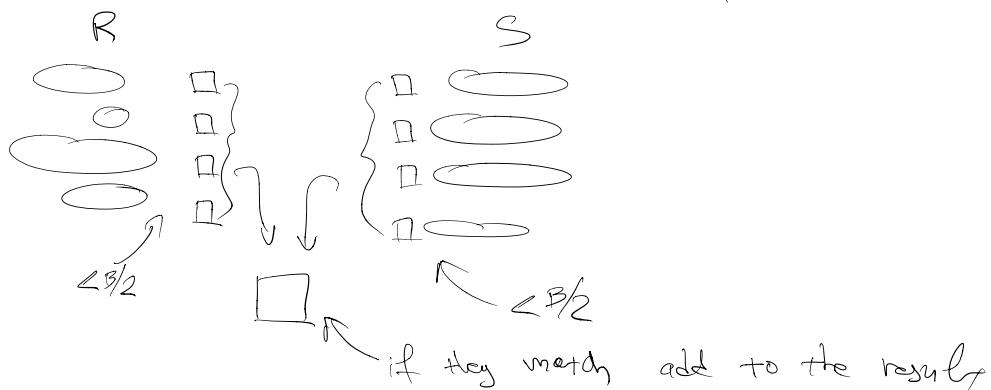
consider using replacement sort

it results to runs with size $\approx 2B$

sorted runs after pass 0 using replacement sort

$$R \rightarrow \frac{M}{2B} < \frac{B}{2} \quad S \rightarrow \frac{N}{2B} < B/2$$

we allocate a buffer per sorted run per file



$$\text{cost} = (M+N) \cdot 3$$

Read R \rightarrow writing $< B/2$ # runs of R $2 \cdot M$

Read S \rightarrow writing $< B/2$ # runs of S $2 \cdot N$

Read R and S and merge on the fly : $M+N$

$$(M+N) \cdot 3 = \boxed{4500} \text{ I/Os} \rightarrow \boxed{95}$$