

CS 347:
Distributed Databases and
Transaction Processing
Notes04: Query Optimization

Hector Garcia-Molina
Zoltan Gyongyi

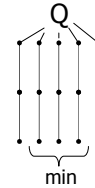
CS 347

Notes 04

1

Query optimization

- Cost estimation
- Strategies for exploring plans



CS 347

Notes 04

2

Cost estimation

- ☞ As in centralized systems:
estimate result sizes

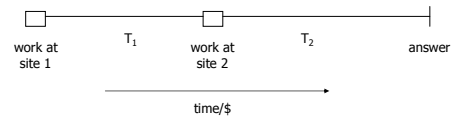
CS 347

Notes 04

3

- ☞ But # of IOs may not be the best metric

E.g., transmission time may
dominate cost



CS 347

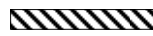
Notes 04

4

Another reason why plain IOs is not enough:




parallelism

Plan A



100 IOs

Plan B

site 1		50 IOs
site 2		70 IOs
site 3		50 IOs

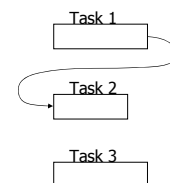
CS 347

Notes 04

5

- Cost metrics
 - IOs, bytes transmitted, \$, ...
 - Can add together

- Response time metric
 - Cannot add
 - Need scheduling and dependency info
 - Skew is important



CS 347

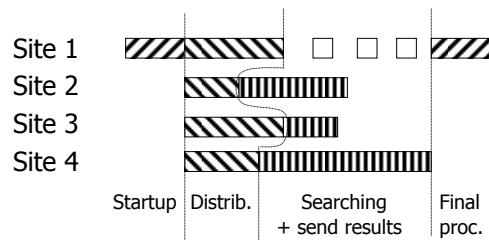
Notes 04

6

⇔ Take into account:
(in parallel/distributed systems)

- Start up costs (for parallel operation)
- Data distribution costs/time
- Contention
 - Memory, disk, network, ...
- Assembling result

Example: Response time



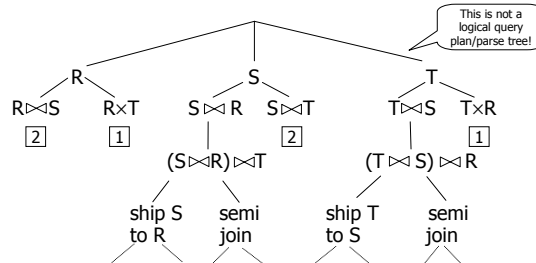
Searching strategies

- 1) Exhaustive (with pruning)
- 2) Hill climbing (greedy)
- 3) Query separation

1) Exhaustive

- Consider "all" query plans with a set of techniques
- Prune some plans
- Heuristics

Example: join $(R \overset{A}{\text{---}} S \overset{B}{\text{---}} T) \mid R \mid > \mid S \mid > \mid T \mid$

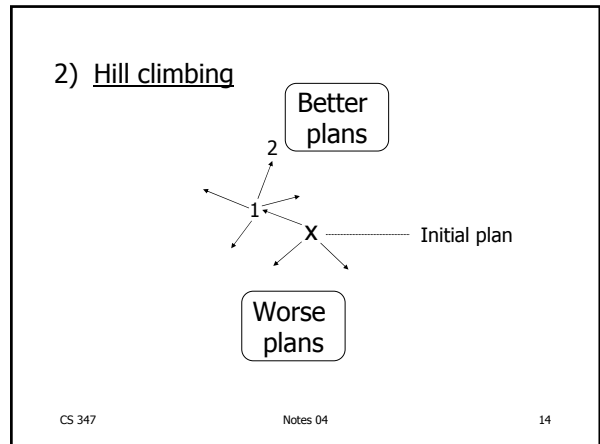
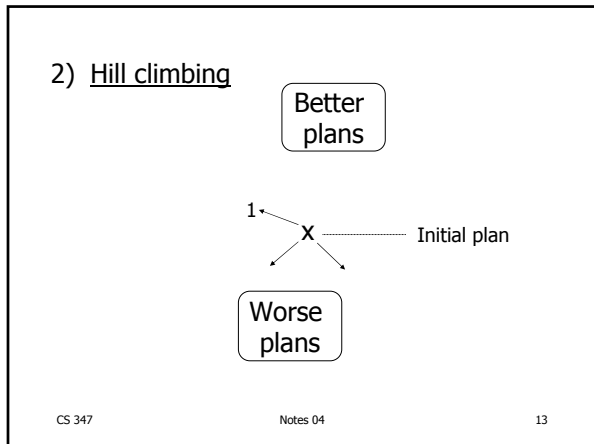


- 1 Prune because cross-product not necessary
- 2 Prune because larger relation first

☞ In generating plans, keep goal in mind:

E.g., goal is parallelism in system with fast net → consider partitioning relations first

E.g., goal is reduction of net traffic → consider semi-joins



Example $R \bowtie S \bowtie T \bowtie V$

Rel	Site	Size
R	1	10
S	2	20
T	3	30
V	4	40

tuple size = 1

Goal: minimize data transmission

CS 347 Notes 04 15

Initial plan: send relations to one site

What site do we send all relations to?

To site 1: cost=20+30+40=90

To site 2: cost=10+30+40=80

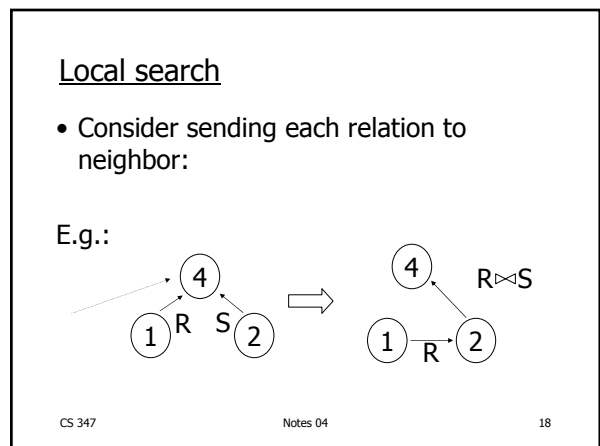
To site 3: cost=10+20+40=70

To site 4: cost=10+20+30=60 📄

CS 347 Notes 04 16

P₀: R (1 → 4)
 S (2 → 4)
 T (3 → 4)
 Compute $R \bowtie S \bowtie T \bowtie V$ at site 4

CS 347 Notes 04 17

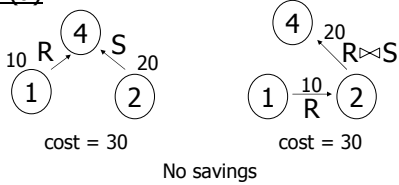


Assume: size $R \bowtie S = 20$

$S \bowtie T = 5$

$T \bowtie V = 1$

Option (a)

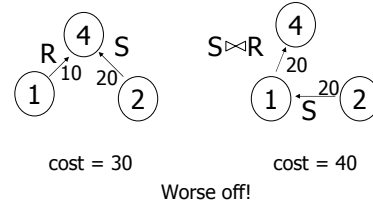


CS 347

Notes 04

19

Option (b)

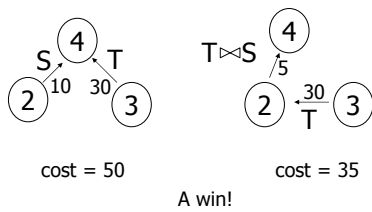


CS 347

Notes 04

20

Option (c)

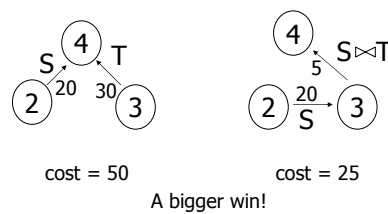


CS 347

Notes 04

21

Option (d)



CS 347

Notes 04

22

P1: P1a: $S(2 \rightarrow 3)$

$\alpha = S \bowtie T$

P1b: $R(1 \rightarrow 4)$

$\alpha(3 \rightarrow 4)$

compute answer at site 4

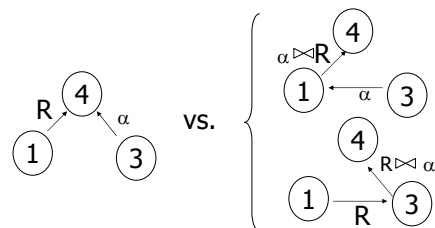
CS 347

Notes 04

23

Repeat local search

Treat $\alpha = S \bowtie T$ as relation



CS 347

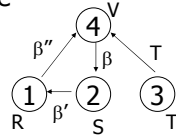
Notes 04

24

Hill climbing may miss best plan!

Example: best plan could be

P_B: T (3 → 4)
 $\beta = T \bowtie V$
 $\beta (4 \rightarrow 2)$
 $\beta' = \beta \bowtie S$
 $\beta' (2 \rightarrow 1)$
 $\beta'' = \beta' \bowtie R$
 [optional] $\beta'' (1 \rightarrow 4)$

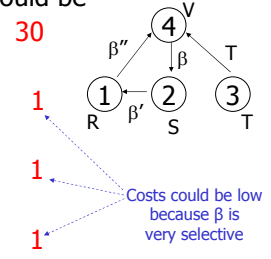


Compute answer

Hill climbing may miss best plan!

Example: best plan could be

P_B: T (3 → 4)
 $\beta = T \bowtie V$
 $\beta (4 \rightarrow 2)$
 $\beta' = \beta \bowtie S$
 $\beta' (2 \rightarrow 1)$
 $\beta'' = \beta' \bowtie R$
 [optional] $\beta'' (1 \rightarrow 4)$



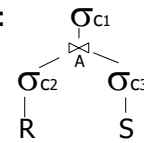
Compute answer **33 = total**

3) Query separation

- Separate query into 2 or more steps
- Optimize each step independently

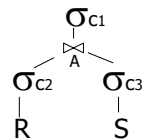
Example: simple queries

E.g.:



1. Compute $R' = \Pi_A[\sigma_{C_2} R]$
 $S' = \Pi_A[\sigma_{C_3} S]$
2. Compute $J = R' \bowtie S'$

1. Compute $R' = \Pi_A[\sigma_{C_2} R]$
 $S' = \Pi_A[\sigma_{C_3} S]$
2. Compute $J = R' \bowtie S'$



3. Compute

$$\text{Ans} = \sigma_{C_1}\{[J \bowtie \sigma_{C_2} R] \bowtie [J \bowtie \sigma_{C_3} S]\}$$

In other words:

- Compute "A" values in answer (steps 1 and 2)
- Get tuples from sites with matching "A" values and compute answer (step 3)

Simple query

- Relations have a single attribute
- Output has a single attribute
E.g., $J \leftarrow R' \bowtie S'$

CS 347

Notes 04

31

Idea

- Decompose query into
 - Local processing
 - Simple query (or queries)
 - Final processing
- Optimize simple query

CS 347

Notes 04

32

Philosophy

- Hard part is distributed join
- Do this part with only keys;
get rest of data later
- Simpler to optimize simple queries

CS 347

Notes 04

33

Summary: Query optimization

- Cost estimation
- Strategies
 - Exhaustive
 - Hill climbing
 - Separation

CS 347

Notes 04

34

Words of wisdom

“Optimization is like chess playing”

That is, may have to make sacrifices
(move data, partition relations, build indexes)
for later gains!

CS 347

Notes 04

35