CSE 544: Optimizations, Size Estimation

Monday, 5/24/2004

Reading

- Book, Chapter 15

Optimizations Based on Semijoins

Semi-join based optimizations

- $R \ltimes S = \Pi_{A1,\ldots,An}(R \bowtie S)$
- Where the schemas are:
  - Input: R(A1,…An), S(B1,…,Bm)
  - Output: T(A1,…,An)

Optimizations Based on Semijoins

Semijoins: a bit of theory (see [AHV])

- Given a conjunctive query:

$Q :- R_1, R_2, \ldots, R_n$

- A full reducer for Q is a program:

$R_{i1} := R_{i1} \ltimes R_{j1}$
$R_{i2} := R_{i2} \ltimes R_{j2}$
$\ldots$
$R_{ip} := R_{ip} \ltimes R_{jp}$

- Such that no dangling tuples remain in any relation

Optimizations Based on Semijoins

- Example: Q :- R1(A,B), R2(B,C), R3(C,D)
- A full reducer is:

R2(B,C) := R2(B,C), R1(A,B)

R3(C,D) := R3(C,D), R2(B,C)

R2(B,C) := R2(B,C), R3(C,D)

R1(A,B) := R1(A,B), R2(B,C)

Optimizations Based on Semijoins

- Example:

Q :- R1(A,B), R2(B,C), R3(A,C)

- Doesn't have a full reducer (we can reduce forever)

Theorem a query has a full reducer iff it is "acyclic"

Optimizations Based on Semijoins

- Semijoins in [Chaudhuri'98]

```
CREATE VIEW DepAvgSal As (
        SELECT E.did, Avg(E.Sal) AS avgsal
        FROM Emp E
        GROUP BY E.did)

SELECT E.eid, E.sal
FROM Emp E, Dept D, DepAvgSal V
WHERE E.did = D.did AND E.did = V.did
        AND E.age < 30 AND D.budget > 100k
        AND E.sal > V.avgsal
```

7

Optimizations Based on Semijoins

- First idea:

```
CREATE VIEW LimitedAvgSal As (
        SELECT E.did, Avg(E.Sal) AS avgsal
        FROM Emp E, Dept D
        WHERE E.did = D.did AND D.buget > 100k
        GROUP BY E.did)

SELECT E.eid, E.sal
FROM Emp E, Dept D, LimitedAvgSal V
WHERE E.did = D.did AND E.did = V.did
        AND E.age < 30 AND D.budget > 100k
        AND E.sal > V.avgsal
```

8

Optimizations Based on Semijoins

- Better: full reducer

```
CREATE VIEW PartialResult AS
        (SELECT E.id, E.sal, E.did
        FROM Emp E, Dept D
        WHERE E.did=D.did AND E.age < 30
        AND D.budget > 100k)

CREATE VIEW Filter AS
        (SELECT DISTINCT P.did FROM PartialResult P)

CREATE VIEW LimitedAvgSal AS
        (SELECT E.did, Avg(E.Sal) AS avgsal
        FROM Emp E, Filter F
        WHERE E.did = F.did GROUP BY E.did)
```

9

Optimizations Based on Semijoins

```
SELECT P.eid, P.sal
FROM PartialResult P, LimitedDepAvgSal V
WHERE P.did = V.did AND P.sal > V.avgsal
```

10

Modern Query Optimizers

- Volcano
  - Rewrite rules
  - Extensible
- Starburst
  - Keeps query blocks
  - Interblock, intrablock optimizations

11

Size Estimation

The problem: Given an expression E, compute T(E) and V(E, A)

- This is hard without computing E
- Will 'estimate' them instead

12

Size Estimation

Estimating the size of a projection

- Easy: $T(\Pi_L(R)) = T(R)$
- This is because a projection doesn't eliminate duplicates

13

Size Estimation

Estimating the size of a selection

- $S = \sigma_{A=c}(R)$
  - T(S) san be anything from 0 to T(R) – V(R,A) + 1
  - Estimate: T(S) = T(R)/V(R,A)
  - When V(R,A) is not available, estimate T(S) = T(R)/10
- $S = \sigma_{A<c}(R)$
  - T(S) can be anything from 0 to T(R)
  - Estimate: T(S) = (c - Low(R, A))/(High(R,A) - Low(R,A))
  - When Low, High unavailable, estimate T(S) = T(R)/3

14

Size Estimation

Estimating the size of a natural join, $R |\times|_A S$

- When the set of A values are disjoint, then $T(R |\times|_A S) = 0$
- When A is a key in S and a foreign key in R, then $T(R |\times|_A S) = T(R)$
- When A has a unique value, the same in R and S, then $T(R |\times|_A S) = T(R)\ T(S)$

15

Size Estimation

Assumptions:

- *Containment of values*: if V(R,A) <= V(S,A), then the set of A values of R is included in the set of A values of S
  - Note: this indeed holds when A is a foreign key in R, and a key in S
- *Preservation of values*: for any other attribute B, $V(R |\times|_A S, B) = V(R, B)$ (or V(S, B))

16

Size Estimation

Assume V(R,A) <= V(S,A)

- Then each tuple t in R joins *some* tuple(s) in S
  - How many ?
  - On average T(S)/V(S,A)
  - t will contribute T(S)/V(S,A) tuples in $R |\times|_A S$
- Hence $T(R |\times|_A S) = T(R)\ T(S) / V(S,A)$

In general: $T(R |\times|_A S) = T(R)\ T(S) / \max(V(R,A),V(S,A))$

17

Size Estimation

Example:

- T(R) = 10000, T(S) = 20000
- V(R,A) = 100, V(S,A) = 200
- How large is $R |\times|_A S$?

Answer: $T(R |\times|_A S) = 10000\ 20000/200 = 1M$

18

Size Estimation

Joins on more than one attribute:

- $T(R \bowtie_{A,B} S) =$

  $T(R)\ T(S)/(\max(V(R,A),V(S,A))*\max(V(R,B),V(S,B)))$

19

Histograms

- Statistics on data maintained by the RDBMS
- Makes size estimation much more accurate (hence, cost estimations are more accurate)

20

Histograms

Employee(ssn, name, salary, phone)

- Maintain a histogram on salary:

Salary:	0..20k	20k..40k	40k..60k	60k..80k	80k..100k	> 100k
Tuples	200	800	5000	12000	6500	500

- T(Employee) = 25000, but now we know the distribution

21

Histograms

Ranks(rankName, salary)

- Estimate the size of Employee $\bowtie_{Salary}$ Ranks

Employee	0..20k	20k..40k	40k..60k	60k..80k	80k..100k	> 100k
	200	800	5000	12000	6500	500

Ranks	0..20k	20k..40k	40k..60k	60k..80k	80k..100k	> 100k
	8	20	40	80	100	2

22

Histograms

- Eqwidth

0..20	20..40	40..60	60..80	80..100
2	104	9739	152	3

- Eqdepth

0..44	44..48	48..50	50..56	55..100
2000	2000	2000	2000	2000

23

End of "Normal" Lectures

- What's next ?
- Two lectures on advanced topics:
  - Queries with uncertainties
  - Security issues in data sharing
- Projects presentations

24