CSE544
Database Statistics

Tuesday, February 15th, 2011

Outline

- Chapter 15 in the textbook

Query Optimization

<u>Three major components</u>:

1. Search space

2. Algorithm for enumerating query plans

3. Cardinality and cost estimation

3. Cardinality and Cost Estimation

- Collect statistical summaries of stored data
- Estimate <u>size</u> (=cardinality) in a bottom-up fashion
  - This is the most difficult part, and still inadequate in today's query optimizers
- Estimate <u>cost</u> by using the estimated size
  - Hand-written formulas, similar to those we used for computing the cost of each physical operator

Statistics on Base Data

- Collected information for each relation
  - Number of tuples (cardinality)
  - Indexes, number of keys in the index
  - Number of physical pages, clustering info
  - Statistical information on attributes
    - Min value, max value, number distinct values
    - Histograms
  - Correlations between columns (hard)
- Collection approach: periodic, using sampling

Size Estimation Problem

S = SELECT list
    FROM R1, …, Rn
    WHERE cond_1 AND cond_2 AND . . . AND cond_k

Given T(R1), T(R2), …, T(Rn)
Estimate T(S)

How can we do this ? Note: doesn't have to be exact.

Size Estimation Problem

```
S = SELECT list
    FROM   R1, …, Rn
    WHERE  cond1 AND cond2 AND . . . AND condk
```

Remark: $T(S) \leq T(R1) \times T(R2) \times \ldots \times T(Rn)$

Selectivity Factor

- Each condition *cond* reduces the size by some factor called *selectivity factor*
- Assuming independence, multiply the selectivity factors

Example

R(A,B)
S(B,C)
T(C,D)

```
SELECT *
FROM R, S, T
WHERE R.B=S.B and S.C=T.C and R.A<40
```

T(R) = 30k, T(S) = 200k, T(T) = 10k

Selectivity of R.B = S.B is 1/3
Selectivity of S.C = T.C is 1/10
Selectivity of R.A < 40 is ½

What is the estimated size of the query output ?

Rule of Thumb

- If selectivities are unknown, then: selectivity factor = 1/10 [System R, 1979]

Using Data Statistics

- Condition is A = c   /* value selection on R */
  - Selectivity = 1/V(R,A)
- Condition is A < c   /* range selection on R */
  - Selectivity = (c - Low(R, A))/(High(R,A) - Low(R,A))T(R)
- Condition is A = B   /* $R \bowtie_{A=B} S$ */
  - Selectivity = 1 / max(V(R,A),V(S,A))
  - (will explain next)

Assumptions

- *Containment of values*: if V(R,A) <= V(S,B), then the set of A values of R is included in the set of B values of S
  - Note: this indeed holds when A is a foreign key in R, and B is a key in S

- *Preservation of values*: for any other attribute C, $V(R \bowtie_{A=B} S, C) = V(R, C)$ (or $V(S, C)$)

Selectivity of R $\bowtie_{A=B}$ S

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

- Each tuple t in R joins with T(S)/V(S,B) tuple(s) in S
- Hence T(R $\bowtie_{A=B}$ S) = T(R) T(S) / V(S,B)

In general: T(R $\bowtie_{A=B}$ S) = T(R) T(S) / max(V(R,A),V(S,B))

Size Estimation for Join

Example:

- $T(R) = 10000$, $T(S) = 20000$
- $V(R,A) = 100$, $V(S,B) = 200$
- How large is $R \bowtie_{A=B} S$?

Histograms

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

Histograms

Employee(ssn, name, age)

T(Employee) = 25000, V(Empolyee, age) = 50
min(age) = 19, max(age) = 68

$\sigma_{age=48}$(Empolyee) = ? $\sigma_{age>28 \text{ and } age<35}$(Empolyee) = ?

Histograms

Employee(ssn, name, age)

T(Employee) = 25000, V(Empolyee, age) = 50
min(age) = 19, max(age) = 68

$\sigma_{age=48}$(Empolyee) = ? $\sigma_{age>28 \text{ and } age<35}$(Empolyee) = ?

Estimate = 25000 / 50 = 500 Estimate = 25000 * 6 / 60 = 2500

Histograms

Employee(<u>ssn</u>, name, age)

T(Employee) = 25000, V(Empolyee, age) = 50
min(age) = 19, max(age) = 68

$\sigma_{\text{age=48}}$(Empolyee) = ? $\sigma_{\text{age>28 and age<35}}$(Empolyee) = ?

Age:	0..20	20..29	30-39	40-49	50-59	> 60
Tuples	200	800	5000	12000	6500	500

Histograms

Employee(ssn, name, age)

T(Employee) = 25000, V(Empolyee, age) = 50
min(age) = 19, max(age) = 68

$\sigma_{age=48}$(Empolyee) = ? $\sigma_{age>28 \text{ and } age<35}$(Empolyee) = ?

Age:	0..20	20..29	30-39	40-49	50-59	> 60
Tuples	200	800	5000	12000	6500	500

Estimate = 1200 Estimate = 2*80 + 5*500 = 2660

Types of Histograms

- How should we determine the bucket boundaries in a histogram ?

Types of Histograms

- How should we determine the bucket boundaries in a histogram ?

- Eq-Width
- Eq-Depth
- Compressed
- V-Optimal histograms

Employee(ssn, name, age)

Histograms

Eq-width:

Age:	0..20	20..29	30-39	40-49	50-59	> 60
Tuples	200	800	5000	12000	6500	500

Eq-depth:

Age:	0..20	20..29	30-39	40-49	50-59	> 60
Tuples	1800	2000	2100	2200	1900	1800

Compressed: store separately highly frequent values: (48,1900)

Some Definitions

- [Ioannidis: *The history of histograms*, 2003]
- Relation R, attribute X
- Value set of R.X is $V = \{v_1 < v_2 < \ldots < v_n\}$
- Spread of v_k is $s_k = v_{k+1} - v_k$
- Frequency f_k = # of tuples with $R.X = v_k$
- Area $a_k = s_k \times f_k$

V-Optimal Histograms

- Defines bucket boundaries in an optimal way, to minimize the error over all point queries
- Computed rather expensively, using dynamic programming
- Modern databases systems use V-optimal histograms or some variations

V-Optimal Histograms

Formally:

- The data distribution of R.X is:
  - $T = \{(v_1,f_1), \ldots, (v_n,f_n)\}$
- A histogram for R.X with β buckets is:
  - $H = \{(u_1,e_1), \ldots, (u_\beta,e_\beta)\}$
- The estimation at point x is: $e(x) = e_i$, where $u_i \leq x < u_{i+1}$

V-Optimal Histograms

Formally:

- The v-optimal histogram with β buckets is the histogram $H = \{(u_1,g_1), \ldots, (u_\beta,f_\beta)\}$ that minimizes:

$$\Sigma_{i=1,n} |e(v_i) - f_i|^2$$

- Can be computed using dynamic programming

Difficult Questions on Histograms

- Small number of buckets
  - Hundreds, or thousands, but not more
  - WHY ?
- *Not* updated during database update, but recomputed periodically
  - WHY ?
- Multidimensional histograms rarely used
  - WHY ?

Summary of Query Optimization

- Three parts:
  – search space, algorithms, size/cost estimation
- Ideal goal: find optimal plan. But
  – Impossible to estimate accurately
  – Impossible to search the entire space
- Goal of today's optimizers:
  – Avoid very bad plans