Optimization: Cost Estimation: Scans

R&G Chapter 15

(slides adapted from content by J.Gehrke, J.Shanmugasundaram, and/or C.Koch)

Reminder: Project 1 Deadline Extended to Tuesday Night

20 bonus points if you've already submitted

Project 2 will be assigned Wednesday

Homework 4 (due Sunday night) will be posted tonight
(12:01 Monday Morning)

Enumerating Possible Plans

- There are many algorithms suitable for processing a specific data set:

1. How do we identify feasible algorithms.
2. How do we choose between them

Single Relation Queries

```
SELECT    O.Rank, COUNT(*),
FROM      Officers O
WHERE     O.Rank >= 2
  AND     O.Age > 20 AND O.Age < 30
GROUP BY  O.Rank
HAVING    COUNT(DISTINCT O.Ship) > 2
```

What is this query asking?

Single Relation Queries

```
SELECT    O.Rank, COUNT(*),
FROM      Officers O
WHERE     O.Rank >= 2
  AND     O.Age > 20 AND O.Age < 30
GROUP BY  O.Rank
HAVING    COUNT(DISTINCT O.Ship) > 2
```

Group Exercise: Compose a (naive) RA expression for this query (using scan, select, project, aggregate)

Single Relation Queries

```
SELECT   O.Rank, cnt,
FROM
  (SELECT O.Rank,
          COUNT(*) AS cnt,
          COUNT(DISTINCT O.Ship) AS dship
   FROM   Officers O
   WHERE    O.Rank >= 2
     AND    O.Age > 20 AND O.Age < 30
  ) O
WHERE    dship > 2
GROUP BY O.Rank
```

Group Exercise: Compose a (naive) RA expression for this query (using scan, select, project, aggregate)

Evaluation Strategies

- How do we read in the data (Which Access Path)?
  - No Index
  - Single Unsorted Index
  - Single Index Sorted on the GB Key
  - Multiple-Index
  - Index-Only Access
- How do we implement the Group-By?
  - Hash-Grouping
  - Sort/Group

Single Unsorted Index Scans have a Fixed Reduction factor. We'll cover this next week
Sorted Index scans allow us to avoid sorting for the aggregate
Multiple-index scans are possible if we have indices that store record IDs. We can scan each index and compute the intersection of all record IDs.
If the index key includes ALL attributes referenced by the query, we can actually walk the index and reconstruct the relevant tuples from the key.


Any Questions?

No Index-Sort/Group

Officers

50 bytes/tuple

80 tuples/page

500 pages

No Index-Sort/Group

1) File Scan Cost: ?

<u>Officers</u>

50 bytes/tuple

80 tuples/page

500 pages

No Index-Sort/Group

1) File Scan Cost: 500 IOs

<u>Officers</u>

50 bytes/tuple

80 tuples/page

500 pages

No Index-Sort/Group

1) File Scan Cost: 500 IOs

2) Select/Project/Write Cost: ?

<u>Officers</u>

50 bytes/tuple

80 tuples/page

500 pages

No Index-Sort/Group

1) File Scan Cost: 500 IOs

2) Select/Project/Write Cost: <500 IOs

<u>Officers</u>
50 bytes/tuple
80 tuples/page
500 pages

Reduction Factor

Selection Eliminates Tuples

Projection Eliminates Columns

Selection/Projection Reduce The Size of a Relation!

Reduction Factor

(The fraction of Tuples/Bytes present in the output)

Officers

50 bytes/tuple

80 tuples/page

500 pages

No Index-Sort/Group

1) File Scan Cost: 500 IOs

2) Select/Project/Write Cost: <500 IOs

Reduction Factor 0.75

<u>Officers</u>
50 bytes/tuple
80 tuples/page
500 pages

No Index-Sort/Group

1) File Scan Cost: 500 IOs

2) Select/Project/Write Cost: 375 IOs

Reduction Factor 0.75

Officers

50 bytes/tuple

80 tuples/page

500 pages

No Index-Sort/Group

1) File Scan	Cost:	500 IOs
2) Select/Project/Write	Cost:	375 IOs

Reduction Factor 0.75

3) Finish the Sort (50 frames)	Cost:	?

<u>Officers</u>
50 bytes/tuple
80 tuples/page
500 pages

No Index-Sort/Group

1) File Scan	Cost:	500 IOs
2) Select/Project/Write	Cost:	375 IOs

Reduction Factor 0.75

3) Finish the Sort (50 frames)	Cost:	375 IOs

<u>Officers</u>
50 bytes/tuple
80 tuples/page
500 pages

No Index-Sort/Group

1) File Scan	Cost:	500 IOs
2) Select/Project/Write	Cost:	375 IOs

Reduction Factor 0.75

3) Finish the Sort (50 frames)	Cost:	375 IOs
4) Scan + Aggregate	Cost:	?

<u>Officers</u>

50 bytes/tuple

80 tuples/page

500 pages

No Index-Sort/Group

1) File Scan   Cost: 500 IOs

2) Select/Project/Write   Cost: 375 IOs

Reduction Factor 0.75

3) Finish the Sort (50 frames)   Cost: 375 IOs

4) Scan + Aggregate   Cost: [included]

<u>Officers</u>
50 bytes/tuple
80 tuples/page
500 pages


Any Questions?

Reduction Factors

- Reduction factors give us a way to estimate how many tuples are in a relation.
- This is crucial for estimating the performance of a query plan.
- Not just in terms of the IO cost of the plan, but also the compute cost.
  - **IO**: #pages $\leftrightarrow$ **Compute**: #tuples

Estimating Reduction Factors

(Selection)

Simplify to Conjunctive Normal Form

$$\sigma_{a_1 \wedge \ldots \wedge a_n}(R) = \sigma_{a_1}(\ldots \sigma_{a_n}(R))$$

Split Conjunctions

Estimating Reduction Factors

(Selection)

Simplify to Conjunctive Normal Form

$$a_1 \vee \ldots \vee a_n = \neg(\neg a_1 \wedge \ldots \neg a_n)$$

(De Morgan's Law)

$$RF(a_1 \vee \ldots \vee a_n) = 1 - ((1 - RF(a_1)) \cdot \ldots \cdot (1 - RF(a_n)))$$

(RF(¬a) = 1-RF(a))


Any Questions?

Estimating Reduction Factors

```
SELECT *
FROM R
WHERE A = 1
```

Assume a Uniform Distribution

$$\frac{1}{NKeys(A)}$$

How do we get NKeys?

Estimating Reduction Factors

```
SELECT *
FROM R, S
WHERE R.A IN (1, 2, …)
```

Assume a Uniform Distribution

$$\frac{\#Values}{NKeys(A)}$$

Estimating Reduction Factors

```
SELECT *
FROM R
WHERE A > 1
```

Assume a Uniform Distribution

$$\frac{High(A) - value}{High(A) - Low(A)}$$

How do we get High/Low?

Estimating Reduction Factors

```
SELECT *
FROM R, S
WHERE R.A = S.B
```

Nested-Loop Index Scan on the Smaller Relation

$$\frac{1}{\max(NKeys(A), NKeys(B))}$$


Any Questions?

Index Scan

1) Index Scan　　　Cost: ? IOs

What Indexes are Available?

O.Age?　　　O.Rank?

20 < Age < 30
RF = ?

Rank > 2
RF = ?

Ranks: 0-5 (in increments of 0.5)

Ages: 15-100 (in increments of 1)

<u>Officers</u>

50 bytes/tuple

80 tuples/page

500 pages

Assuming a uniform distribution, 85 Age keys, 11 Rank keys
9 legitimate ages, 9 / 85 = 0.105
6 legitimate ranks, 6 / 11 = 0.545
Use the Age index!

Index Scan-Sort/Group

1) Index Scan Cost: ~53 IOs

2) Select/Project/Write Cost: ~23 IOs

Assume Projection has an RF of 0.8

Ranks: 0-5 (in increments of 0.5)

Ages: 15-100 (in increments of 1)

<u>Officers</u>

50 bytes/tuple

80 tuples/page

500 pages

We still get the 0.545 reduction factor on the 2nd select
Assume another 0.8 on the projection
Total = 0.4346

Index Scan-Sort/Group

1) Index Scan Cost: ~53 IOs

2) Select/Project~~/Write~~ Cost: ~~23 IOs~~

Assume Projection has an RF of 0.8

Ranks: 0-5 (in increments of 0.5)

Ages: 15-100 (in increments of 1)

Officers

50 bytes/tuple

80 tuples/page

500 pages

We still get the 0.545 reduction factor on the 2nd select
Assume another 0.8 on the projection
Total = 0.4346

Index Scan-Sort/Group

1) Index Scan Cost: ~53 IOs

2) Select/Project~~/Write~~ Cost: ~~23 IOs~~

Assume Projection has an RF of 0.8

3) In-Mem Sort (50 frames) Cost: 0 IOs

4) Scan + Aggregate Cost: 0 IOs

Ranks: 0-5 (in increments of 0.5)

Ages: 15-100 (in increments of 1)

<u>Officers</u>

50 bytes/tuple

80 tuples/page

500 pages

We still get the 0.545 reduction factor on the 2nd select
Assume another 0.8 on the projection
Total = 0.4346


Any Questions?