Introduction to Database Systems
CSE 344

Lecture 11:
Basics of Query Optimization and
Query Cost Estimation
Announcements

• HW3 Azure V12 upgrade

• Section attendance
Review

• What is a disk block? (aka page)

• What is an index?
  – What data structures are used to represent indexes in memory?

• What are clustered/unclustered indexes?
Recap – Indexes

V(M, N, P);

SELECT *
FROM V
WHERE V.M = 33

Suppose we only had one of these indexes. How can the optimizer use it?

INDEX I1 on V(M)
INDEX I2 on V(M,P)
INDEX I3 on V(P,M)
Two typical kinds of queries

- **Point queries**
  - What data structure should be used for index?

- **Range queries**
  - What data structure should be used for index?

```sql
SELECT * 
FROM Movie 
WHERE year = ?
```

```sql
SELECT * 
FROM Movie 
WHERE year >= ? AND year <= ?
```
Basic Index Selection Guidelines

- Consider queries in workload in order of importance

- Consider relations accessed by query
  - No point indexing other relations

- Look at WHERE clause for possible search key

- Consider how each query will be processed
  - Which predicate will be processed first?

- Try to choose indexes that speed-up multiple queries
SELECT *  
FROM R  
WHERE K>? and K<?
SELECT * 
FROM R 
WHERE K>? and K<?
SELECT * 
FROM R 
WHERE K>? and K<?
SELECT * 
FROM R 
WHERE K>? and K<?
Cost Models

- Cost of reading from disk
- Cost of single operators
- Cost of query plans
Cost of Reading
Data From Disk
Cost Parameters

• Cost = I/O + CPU + Network BW
  – We will focus on I/O in this class

• Parameters:
  – $B(R) = \#\ of\ blocks\ (i.e.,\ pages)\ for\ relation\ R$
  – $T(R) = \#\ of\ tuples\ in\ relation\ R$
  – $V(R, a) = \#\ of\ distinct\ values\ of\ attribute\ a$
    • When $a$ is a key, $V(R,a) = T(R)$
    • When $a$ is not a key, $V(R,a)$ can be anything $< T(R)$

• Where do these values come from?
  – DBMS collects statistics about data on disk
Selectivity Factors for Conditions

- $A = c$  /* $\sigma_{A=c}(R)$ */
  - Selectivity = $1/V(R,A)$

- $A < c$  /* $\sigma_{A<c}(R)$ */
  - Selectivity = $(c - \min(R, A))/\left(\max(R,A) - \min(R,A)\right)$

- $c_1 < A < c_2$  /* $\sigma_{c_1<A<c_2}(R)$ */
  - Selectivity = $(c_2 - c_1)/\left(\max(R,A) - \min(R,A)\right)$
Cost of Reading Data From Disk

- Sequential scan for relation R costs $B(R)$

- Index-based selection
  - Estimate selectivity factor $X$ (see previous slide)
  - Clustered index: $X^*B(R)$
  - Unclustered index $X^*T(R)$

Note: we ignore I/O cost for index pages
Index Based Selection

- Example:
  \[ B(R) = 2000 \]
  \[ T(R) = 100,000 \]
  \[ V(R, a) = 20 \]

- Table scan:
- Index based selection:

  \[ \text{cost of } \sigma_{a=v}(R) = ? \]
Index Based Selection

- Example:
  - Table scan: $B(R) = 2000$ I/Os
  - Index based selection:

<table>
<thead>
<tr>
<th>$B(R)$</th>
<th>$T(R)$</th>
<th>$V(R, a)$</th>
</tr>
</thead>
<tbody>
<tr>
<td>2000</td>
<td>100,000</td>
<td>20</td>
</tr>
</tbody>
</table>

$cost \ of \ \sigma_{a=v}(R) = ?$
Index Based Selection

• Example:
  
  \[
  \begin{array}{l}
  B(R) = 2000 \\
  T(R) = 100,000 \\
  V(R, a) = 20 \\
  \end{array}
  \]

• Table scan: \( B(R) = 2,000 \) I/Os

• Index based selection:
  
  – If index is clustered:
  
  – If index is unclustered:

  \text{cost of } \sigma_{a=v}(R) = ?
Index Based Selection

- Example:
  
  \[
  \begin{align*}
  B(R) &= 2000 \\
  T(R) &= 100,000 \\
  V(R, a) &= 20
  \end{align*}
  \]

- Table scan: \(B(R) = 2,000 \text{ I/Os}\)

- Index based selection:
  - If index is clustered: \(B(R) \times 1/V(R,a) = 100 \text{ I/Os}\)
  - If index is unclustered:

\[
\text{cost of } \sigma_{a=v}(R) = ?
\]
Index Based Selection

- Example:
  
  | B(R) = 2000 |
  | T(R) = 100,000 |
  | V(R, a) = 20 |

- Table scan: \( B(R) = 2,000 \) I/Os

- Index based selection:
  - If index is clustered: \( B(R) \times \frac{1}{V(R,a)} = 100 \) I/Os
  - If index is unclustered: \( T(R) \times \frac{1}{V(R,a)} = 5,000 \) I/Os

Cost of \( \sigma_{a=v}(R) = ? \)
Index Based Selection

- Example:
  - B(R) = 2000
  - T(R) = 100,000
  - V(R, a) = 20

- Table scan: B(R) = 2,000 I/Os
- Index based selection:
  - If index is clustered: B(R) * 1/V(R,a) = 100 I/Os
  - If index is unclustered: T(R) * 1/V(R,a) = 5,000 I/Os

Lesson: Don’t build unclustered indexes when V(R,a) is small!
Cost of Executing Operators (Focus on Joins)
Outline

• **Join operator algorithms**
  – One-pass algorithms (Sec. 15.2 and 15.3)
  – Index-based algorithms (Sec 15.6)

• **Note about readings:**
  – In class, we discuss only algorithms for joins
  – Other operators are easier: read the book
Join Algorithms

- Hash join
- Nested loop join
- Sort-merge join
Hash Join

Hash join:  \( R \bowtie S \)

• Scan \( R \), build buckets in main memory
• Then scan \( S \) and join
• Cost: \( B(R) + B(S) \)
• Which relation to build the hash table on?

• One-pass algorithm when \( B(R) \leq M \)
  – \( M = \) number of memory pages available
Hash Join Example

Patient(pid, name, address)
Insurance(pid, provider, policy_nb)

Patient $\bowtie$ Insurance

<table>
<thead>
<tr>
<th>Patient</th>
<th>Insurance</th>
</tr>
</thead>
<tbody>
<tr>
<td>1 'Bob'</td>
<td>2 'Blue'</td>
</tr>
<tr>
<td>2 'Ela'</td>
<td>4 'Prem'</td>
</tr>
<tr>
<td>3 'Jill'</td>
<td>4 'Prem'</td>
</tr>
<tr>
<td>4 'Joe'</td>
<td>3 'GrpH'</td>
</tr>
</tbody>
</table>

Two tuples per page
Hash Join Example

Patient \Join Insurance

Memory M = 21 pages

Disk

Patient | Insurance
---|---
1 2 | 2 4 6 6
3 4 | 4 3 1 3
9 6 | 2 8
8 5 | 8 9

Showing pid only

This is one page with two tuples

Some large-enough #
Hash Join Example

Step 1: Scan Patient and **build** hash table in memory
Can be done in method open()

Memory M = 21 pages
Hash h: pid % 5

Disk

<table>
<thead>
<tr>
<th>Patient</th>
<th>Insurance</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>2 4</td>
</tr>
<tr>
<td>3</td>
<td>4 3</td>
</tr>
<tr>
<td>9</td>
<td>2 8</td>
</tr>
<tr>
<td>8</td>
<td>8 9</td>
</tr>
</tbody>
</table>

Input buffer
Hash Join Example

Step 2: Scan Insurance and probe into hash table
Done during calls to next()

Memory M = 21 pages
Hash h: pid % 5

Write to disk or pass to next operator
Hash Join Example

Step 2: Scan Insurance and probe into hash table

Done during calls to next()

Memory $M = 21$ pages

Hash $h$: pid $\% 5$

<p>| | | | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>5</td>
<td>1</td>
<td>6</td>
<td>2</td>
</tr>
<tr>
<td>3</td>
<td>8</td>
<td>4</td>
<td>9</td>
</tr>
</tbody>
</table>

Disk

Patient | Insurance
-------|-----------
1 2     | 2 4
3 4     | 4 3
9 6     | 2 8
8 5     | 8 9

Input buffer

Output buffer

$2 4$
Hash Join Example

Step 2: Scan Insurance and probe into hash table
Done during calls to next()

Memory M = 21 pages
Hash h: pid % 5

Patient | Insurance
---|---
1 2 | 2 4
3 4 | 4 3
9 6 | 2 8
8 5 | 8 9

Input buffer: 4 3
Output buffer: 4 4

Keep going until read all of Insurance

Cost: B(R) + B(S)
Nested Loop Joins

• Tuple-based nested loop $R \bowtie S$
• $R$ is the outer relation, $S$ is the inner relation

```plaintext
for each tuple $t_1$ in $R$ do
  for each tuple $t_2$ in $S$ do
    if $t_1$ and $t_2$ join then output ($t_1$, $t_2$)
```

What is the Cost?
Nested Loop Joins

- Tuple-based nested loop $R \bowtie S$
- $R$ is the outer relation, $S$ is the inner relation

```
for each tuple $t_1$ in $R$ do
  for each tuple $t_2$ in $S$ do
    if $t_1$ and $t_2$ join then output ($t_1$, $t_2$)
```

- **Cost**: $B(R) + T(R) B(S)$
- Multiple-pass since $S$ is read many times

What is the Cost?
Page-at-a-time Refinement

for each page of tuples r in R do
    for each page of tuples s in S do
        for all pairs of tuples t₁ in r, t₂ in s
            if t₁ and t₂ join then output (t₁,t₂)

• Cost: B(R) + B(R)B(S)

What is the Cost?
Page-at-a-time Refinement

Disk

Patient
1 2
3 4
9 6
8 5

Insurance
2 4
4 3
2 8
8 9

Input buffer for Patient
1 2

Input buffer for Insurance
2 4

Output buffer
2 2
Page-at-a-time Refinement

Disk

<table>
<thead>
<tr>
<th>Patient</th>
<th>Insurance</th>
</tr>
</thead>
<tbody>
<tr>
<td>1 2</td>
<td>2 4</td>
</tr>
<tr>
<td>3 4</td>
<td>4 3</td>
</tr>
<tr>
<td>9 6</td>
<td>2 8</td>
</tr>
<tr>
<td>8 5</td>
<td>8 9</td>
</tr>
</tbody>
</table>

Input buffer for Patient:

- 1 2

Input buffer for Insurance:

- 4 3

Output buffer:

- 6 6
- 1 3

CSE 344 - Fall 2016
Page-at-a-time Refinement

Disk

Patient

<table>
<thead>
<tr>
<th>1</th>
<th>2</th>
</tr>
</thead>
<tbody>
<tr>
<td>3</td>
<td>4</td>
</tr>
<tr>
<td>9</td>
<td>6</td>
</tr>
<tr>
<td>8</td>
<td>5</td>
</tr>
</tbody>
</table>

Insurance

<table>
<thead>
<tr>
<th>2</th>
<th>4</th>
</tr>
</thead>
<tbody>
<tr>
<td>4</td>
<td>3</td>
</tr>
<tr>
<td>2</td>
<td>8</td>
</tr>
<tr>
<td>8</td>
<td>9</td>
</tr>
</tbody>
</table>

Input buffer for Patient

1 2

Input buffer for Insurance

2 8

Output buffer

2 2

Keep going until read all of Insurance

Then repeat for next page of Patient… until end of Patient

Cost: $B(R) + B(R)B(S)$
Block-Nested-Loop Refinement

for each group of M-1 pages r in R do
    for each page of tuples s in S do
        for all pairs of tuples t₁ in r, t₂ in s
            if t₁ and t₂ join then output (t₁, t₂)

• Cost: B(R) + B(R)B(S)/(M-1)

What is the Cost?