Query Optimization
Exercise Session 3

Andrey Gubichev

May 5, 2014
select *
from lineitem l, orders o, customers c
where l.l_orderkey=o.o_orderkey
    and o.o_custkey=c.c_custkey
    and c.c_name='Customer#000014993'.

Homework: Task 1
We know $|R_1|$, $|R_2|$, domains of $R_1.x$, $R_2.y$, 

Homework: Task 2
We know $|R1|$, $|R2|$, domains of $R1.x$, $R2.y$, (that is, $|R1.x|$, $|R2.y|$), and whether $x$ and $y$ are keys or not. The selectivity of $\sigma_{R1.x=c}$ can be estimated as...

- if $x$ is the key:
We know $|R_1|$, $|R_2|$, domains of $R_1.x$, $R_2.y$, (that is, $|R_1.x|$, $|R_2.y|$), and whether $x$ and $y$ are keys or not. The selectivity of $\sigma_{R_1.x=c}$ can be estimated as...

- if $x$ is the key: $\frac{1}{|R_1|}$
We know $|R_1|$, $|R_2|$, domains of $R_1.x$, $R_2.y$, (that is, $|R_1.x|$, $|R_2.y|$), and whether $x$ and $y$ are keys or not. The selectivity of $\sigma_{R_1.x=c}$ can be estimated as...

- if $x$ is the key: $\frac{1}{|R_1|}$
- if $x$ is not the key:
We know $|R1|$, $|R2|$, domains of $R1.x$, $R2.y$, (that is, $|R1.x|$, $|R2.y|$), and whether $x$ and $y$ are keys or not. The selectivity of $\sigma_{R1.x=c}$ can be estimated as...

- if $x$ is the key: $\frac{1}{|R1|}$
- if $x$ is not the key: $\frac{1}{|R1.x|}$
We know $|R_1|$, $|R_2|$, $|R_1.x|$, $|R_2.y|$, and whether $x$ and $y$ are keys or not.
First, the size of $R_1 \times R_2$ is
We know $|R_1|$, $|R_2|$, $|R_1.x|$, $|R_2.y|$, and whether $x$ and $y$ are keys or not.
First, the size of $R_1 \times R_2$ is $|R_1||R_2|$
The selectivity of $\Join_{R_1.x=R_2.y}$ can be estimated as...
  ▶ if both $x$ and $y$ are the keys:
We know \(|R_1|, |R_2|, |R_1.x|, |R_2.y|, and whether \(x\) and \(y\) are keys or not.

First, the size of \(R_1 \times R_2\) is \(|R_1| \cdot |R_2|\)

The selectivity of \(\bowtie_{R_1.x = R_2.y}\) can be estimated as...

- if both \(x\) and \(y\) are the keys: \(\frac{1}{\max(|R_1|, |R_2|)}\)
Homework: Task 2

We know $|R_1|$, $|R_2|$, $|R_1.x|$, $|R_2.y|$, and whether $x$ and $y$ are keys or not.
First, the size of $R_1 \times R_2$ is $|R_1||R_2|
The selectivity of $\bowtie_{R_1.x=R_2.y}$ can be estimated as...
  ▶ if both $x$ and $y$ are the keys: $\frac{1}{\max(|R_1|,|R_2|)}$
  ▶ if only $x$ is the key:
We know $|R1|$, $|R2|$, $|R1.x|$, $|R2.y|$, and whether $x$ and $y$ are keys or not.

First, the size of $R1 \times R2$ is $|R1||R2|

The selectivity of $\bowtie_{R1.x=R2.y}$ can be estimated as...

- if both $x$ and $y$ are the keys: $\frac{1}{\max(|R1|,|R2|)}$
- if only $x$ is the key: $\frac{1}{|R1|}$
We know $|R_1|$, $|R_2|$, $|R_1.x|$, $|R_2.y|$, and whether $x$ and $y$ are keys or not.
First, the size of $R_1 \times R_2$ is $|R_1||R_2|
The selectivity of $\Join_{R_1.x=R_2.y}$ can be estimated as...
- if both $x$ and $y$ are the keys: $\frac{1}{\max(|R_1|,|R_2|)}$
- if only $x$ is the key: $\frac{1}{|R_1|}$
- if both $x$ and $y$ are not the keys:
We know $|R_1|$, $|R_2|$, $|R_1.x|$, $|R_2.y|$, and whether $x$ and $y$ are keys or not.

First, the size of $R_1 \times R_2$ is $|R_1||R_2|$

The selectivity of $\Join_{R_1.x=R_2.y}$ can be estimated as...

- if both $x$ and $y$ are the keys: $\frac{1}{\max(|R_1|,|R_2|)}$
- if only $x$ is the key: $\frac{1}{|R_1|}$
- if both $x$ and $y$ are not the keys: $\frac{1}{\max(|R_1.x|,|R_2.y|)}$
Selectivity estimation

We know $|R1|$, $\max(R1.x)$, $\min(R1.x)$, $R1.x$ is arithmetic.

The selectivity of $\sigma_{R1.x>c}$ is
Selectivity estimation

We know $|R1|$, $\max(R1.x)$, $\min(R1.x)$, $R1.x$ is arithmetic.

The selectivity of $\sigma_{R1.x>c}$ is $\frac{\max(R1.x) - c}{\max(R1.x) - \min(R1.x)}$
Selectivity estimation

We know $|R_1|$, $\max(R_1.x)$, $\min(R_1.x)$, $R_1.x$ is arithmetic.

The selectivity of $\sigma_{R_1.x > c}$ is $\frac{\max(R_1.x) - c}{\max(R_1.x) - \min(R_1.x)}$

The selectivity of $\sigma_{c_1 < R_1.x < c_2}$ is
Selectivity estimation

We know $|R_1|$, $\max(R_1.x)$, $\min(R_1.x)$, $R_1.x$ is arithmetic.

The selectivity of $\sigma_{R_1.x>c}$ is $\frac{\max(R_1.x)-c}{\max(R_1.x)-\min(R_1.x)}$

The selectivity of $\sigma_{c_1<R_1.x<c_2}$ is $\frac{c_2-c_1}{\max-\min}$
Homework: Task 3

- $|R| = 1,000$ pages, $|S| = 100,000$ pages
- 1 page - 50 tuples, 1 block - 100 pages
- avg. access = 10 ms, transfer speed = 10,000 pages/sec
- Time for block-nested loops join = ?
Homework: Task 3

- $|R| = 1,000$ pages, $|S| = 100,000$ pages
- 1 page - 50 tuples, 1 block - 100 pages
- avg. access = 10 ms, transfer speed = 10,000 pages/sec
- Time for block-nested loops join = ?
- choose left argument: $R$ vs. $S$, $\frac{1,000}{100}$ vs. $\frac{100,000}{100} \Rightarrow R$
Homework: Task 3

- Time to read one block:
  \[ T_b = \text{avg.seek} + \left(100 \frac{1}{\text{transfer speed}}\right) = 0.02s \]
Homework: Task 3

- Time to read one block:
  \[ T_b = \text{avg.seek} + \left(100 \frac{1}{\text{transfer speed}}\right) = 0.02s \]

- Read 1 block from \( R \), join it with \( S \):
  \[ T_b + \text{time to read } S \approx 10s \]
Homework: Task 3

- Time to read one block:
  \[ T_b = \text{avg. seek} + (100 \frac{1}{\text{transfer speed}}) = 0.02s \]

- Read 1 block from \( R \), join it with \( S \):
  \[ T_b + \text{time to read } S \approx 10s \]

- Repeat it for every block in \( R \):
  \[ T_{BNLJ} = \frac{\# \text{pages in } R}{\text{block size}} (10s) \approx 100s \]
Query Graphs

```sql
select v.titel
from Vorlesungen v, Professoren p
where v.gelesenvon = p.persnr
  and p.name = 'Kant'
  and v.sws = 2;
```
select r.a, s.c
from R r, S s, T t, U u
where r.a = s.a
   and r.b = t.b
   and r.b = u.b;
select r.a, s.c
from R r, S s
where r.a + s.a = 7;
select r.a, s.c
    from R r, S s, T t, U u
    where (r.a + s.b) = (t.b + u.a);
Search space

Search space is defined by:

- Query graph type
Search space

Search space is defined by:

- Query graph type (chain, star, tree, clique, cycle, grid)
- Join tree class
Search space

Search space is defined by:

- Query graph type (chain, star, tree, clique, cycle, grid)
- Join tree class (left-deep, zig-zag, bushy)
- Cost function class
select *
from R1, R2, R3, R4
where R1.a = R2.b
    and R2.c = R3.d
    and R3.e = R4.f

What kind of query graph is it?
Search space

```
select *
from R1, R2, R3, R4
where R1.a = R2.b
    and R2.c = R3.d
    and R3.e = R4.f
```

➤ What kind of query graph is it?
➤ Let’s allow cross-products ⇒ the shape of the query graph does not matter
select *
from R1, R2, R3, R4
where R1.a = R2.b
  and R2.c = R3.d
  and R3.e = R4.f

What kind of query graph is it?
Let’s allow cross-products ⇒ the shape of the query graph does not matter
Count left-deep trees
Count zig-zag trees
Count bushy trees
Roadmap

Good optimizer deals with the following issues:

▶ Cost Model
  ▶ Cost Function
  ▶ Selectivity estimation, statistics
▶ Logical Optimization
  ▶ Search Space
  ▶ Algorithms for Optimal Plan finding
▶ Physical Optimization
  ▶ Enhancing the logical plan with physical operators
DB design

- RTS (Runtime System) – TinyDB
  - how the database is organized on disk? (buffer manager, segments, etc)
  - access methods, operators (scans, joins)
- CTS (Compile-time System) – Goal of the programming exercises
  - Parser (SQLLexer, SQLParser)
  - Semantic Analysis – construct the Query graph, also other transformations
  - Plan generator – logical optimization (join ordering algorithms)
  - Code generator – generates the plan that can be executed by RTS
Selectivity estimation continues...

▶ Our estimations (prev.homework) are far from perfect

▶ Construct specific examples (database schema, concrete instances of relations and selections/joins), where our estimations are very ”bad”

▶ ”Bad” – means that for some queries (give examples of SQL queries) the logical plan will be suboptimal (w.r.t $C_{out}$), if we use these estimations

▶ In other words, bad estimations mislead the optimizer and it outputs a clearly suboptimal plan

▶ Two examples (one for selections, one for joins)
Give an example query instance where the optimal join tree (using $C_{out}$) is bushy and includes a cross product.

Note: the query graph should be connected!
Homework: Task 3 (15 points)

- Using the program from the first exercise as a basis, implement a program that
  - parses SQL queries
  - translates them into tinydb execution plans
  - and executes the query.

- Note: a canonical translation of the joins is fine, but push all predicates of the form `attr = const` down to the base relations
- Don’t do semantic analysis
- Logical optimizer: just takes canonical translation and push down selections, no join ordering
Info

- Slides and exercises: www3.in.tum.de/teaching/ss14/queryopt
- Send any comments, questions, solutions for the exercises etc. to Andrey.Gubichev@in.tum.de
- Exercises due: 9 AM, May 12, 2014