CSC 370

Activity Worksheet:
Physical Query Plans

Dr. Sean Chester

Fall 2022

Notes

In this worksheet, you will practice converting logical query plans into physical query plans. The questions are split into two sections: one in which you apply algebraic laws for cost estimation and one in which you select between two logical query plans based on estimated cost. The first question of each section has been answered already as an exemplar.

Questions

1. Cost Estimation: You have collected the following statistics about tables in your database:

   ```
   T(R)=10000
   T(S)=500
   V(R,x)=1000
   V(R,y)=40
   V(S,x)=200
   V(S,y)=100
   ```

   What is the estimated output size of the operator shown in relational algebra below?

   $$T = R \bowtie_{R.x=S.x \,\wedge\, R.y=S.y} S$$

   Solution:

   For a join, we estimate the probability of the predicate under the assumption of containment of values and multiple that probability by the size of the cross product. Containment of values implies that any tuple in the relation with a smaller domain will match a tuple in the relation with a larger domain. Thus, we have:

   $$\begin{aligned} T(T) &= \frac{T(R) * T(S)}{max(V(R,x), V(S,x)) * max(V(R,y), V(S,y))} \\ &= \frac{10000 * 5}{max(1000, 200) * max(40, 100)} \\ &= \frac{5000000}{1000 * 100} \\ &= 50. \end{aligned}$$

2. Cost Estimation: You have collected the following statistics about tables in your database:

   T(R)=10000
   T(S)=500
   V(R, x)=1000
   V(R, y)=40
   V(R, z)=500
   V(S, x)=200
   V(S, y)=100
   V(S, z)=100

   What is the estimated output size of the operator shown in relational algebra below?

$$T = R \bowtie_{(R.x=S.x \wedge R.y=S.y) \vee R.z=S.z} S$$

   Solution:

3. Cost Estimation: You have collected the following statistics about tables in your database:

   T(R)=10000
   V(R, x)=1000
   V(R, y)=40
   V(R, z)=500

   What is the estimated output size of the operator shown in relational algebra below?

$$S = \gamma_{R.x,R.y,MAX(R.v)\rightarrow topv}(R)$$

   Solution:

4. Cost Estimation: You have collected the following statistics about tables in your database:

T(R)=10000
T(S)=500
V(R,x)=1000
V(S,x)=200

What is the estimated output size of the entire relational algebra expression given below?

$$T = \delta(\pi_x(S) \setminus \pi_y(R))$$

Solution:

5. Generating Alternative Plans: You have collected the following statistics about tables in your database:

T(R)=10000
T(S)=500
V(R,x)=1000
V(S,x)=200

You are given two, equivalent logical query plans below. Showing your work, use cost estimation to determine which query plan has a smaller total I/O cost. Assume that you will use table scans at the leaves; i.e., the cost of reading the tables is the same in both plans and can be cancelled out on both sides of the comparison.

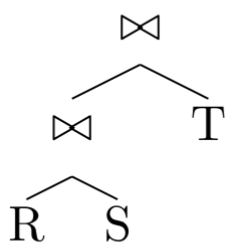

Figure 1: Plan 01

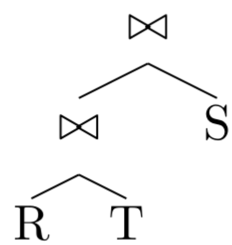

Figure 2: Plan 02

Solution:

We begin by ascribing costs to the first joins, i.e., those closest to the leaves. Note that the natural join will use the attributes in common, which is x in Plan 1 and y in Plan 2. We also need to track the number of y values in the result using the assumption of preservation of values.

Plan 01 We begin...

$$T(R \bowtie S) = \frac{T(R) * T(S)}{max(V(R,x), V(S,x))} = \frac{10000 * 500}{max(1000, 200)} = 5000.$$
$$V(R \bowtie S, y) = V(R, y) = 10.$$

Next, we compute the estimated size of the second join:

$$T((R \bowtie S) \bowtie T) = \frac{T(R \bowtie S) * T(T)}{max(V(R \bowtie S, y), V(T, y))} = \frac{5000 * 2000}{max(10, 100)} = 100000.$$

Thus, the total number of tuples emitted with this logical query plan is 5000+100000 = 105000.

(We have no information about block size nor tuple size here; so, we will have to assume each tuple corresponds to a single I/O, which probably suffices for comparison at this level of detail, anyway.)

Plan 02 We begin again near the leaves, but this time joining R.y and T.y:

$$T(R \bowtie T) = \frac{T(R) * T(T)}{max(V(R,y), V(T,y))} = \frac{10000 * 2000}{max(10, 100)} = 200000.$$
$$V(R \bowtie T, x) = V(R,x) = 1000.$$

Next, we compute the estimated size of the second join:

$$T((R \bowtie T) \bowtie S) = \frac{T(R \bowtie T) * T(S)}{max(V(R \bowtie T, x), V(S,x))} = \frac{200000 * 500}{max(1000, 200)} = 100000.$$

Thus, the total number of tuples emitted with this logical query plan is 200000+100000 = 300000.

Comparison Clearly, Plan 1 is superior, as it emits just over 1/3 as many tuples as Plan 2. We note a couple other observations:

- The final operator had the same estimated cost in both plans. We would expect this, because two equivalent plans should produce the same result. This is one way to check one's work.
- The first join determined the difference between the plans. As a rule of thumb, we would want to minimise result size as much as possible as early as possible.

6. Generating Alternative Plans: You have collected the following statistics about tables in your database:

   ```
   T(R)=10000
   V(R,x)=1000
   ```

   You are given two, equivalent logical query plans below. Showing your work, use cost estimation to determine which query plan has a smaller total I/O cost. Assume that you will use table scans at the leaves; i.e., the cost of reading the tables is the same in both plans and can be cancelled out on both sides of the comparison.

$$\sigma_{x=10 \vee x=42 \vee x=87}$$
$$|$$
$$\delta$$
$$|$$
$$\pi_x$$
$$|$$
$$\mathrm{R}$$

Figure 3: Plan 01

$$\delta$$
$$|$$
$$\pi_x$$
$$|$$
$$\sigma_{x=10 \vee x=42 \vee x=87}$$
$$|$$
$$\mathrm{R}$$

Figure 4: Plan 02

Solution:

Solutions

Question 1

For a join, we estimate the probability of the predicate under the assumption of containment of values and multiple that probability by the size of the cross product. Containment of values implies that any tuple in the relation with a smaller domain will match a tuple in the relation with a larger domain. Thus, we have:

$$
\begin{aligned}
T(T) &= \frac{T(R) * T(S)}{max(V(R,x), V(S,x)) * max(V(R,y), V(S,y))} \\
&= \frac{10000 * 5}{max(1000, 200) * max(40, 100)} \\
&= \frac{5000000}{1000 * 100} \\
&= 50.
\end{aligned}
$$

Question 2

For a join, we estimate the probability of the predicate under the assumption of containment of values and multiple that probability by the size of the cross product. This predicate additionally involves a disjunction, which we treat as an independent event. Thus, we have:

$$
\begin{aligned}
T(T) &= T(R) * T(S) * (1 - (1 - \frac{1}{max(V(R,x), V(S,x)) * max(V(R,y), V(S,y))})(1 - \frac{1}{max(V(R,z), V(S,z))})) \\
&= 10000 * 500 * (1 - (1 - \frac{1}{1000 * 100})(1 - \frac{1}{500})) \\
&= 10050.
\end{aligned}
$$

Question 3

For aggregation, we have one result tuple per group. Here, we pessimistically assume that all groups will have at least one tuple. Therefore:

$$
\begin{aligned}
T(S) &= V(R,x) * V(R,y) \\
&= 1000 * 40 \\
&= 40000.
\end{aligned}
$$

Question 4

We consider each operator in turn:

- The duplicate elimination operator (like the aggregation operator in Question 3) reduces the number of tuples to the number of values.
- The projection operator, on its own, does not affect the number of tuples (because an RDBMS permits bag semantics).
- The difference operator does not make use of containment of values. We take the midpoint of the possible range.

Therefore:

$$\begin{aligned} T(T) &= 0.5 * V(S, x) \\ &= 0.5 * 200 \\ &= 100. \end{aligned}$$

Question 5

We begin by ascribing costs to the first joins, i.e., those closest to the leaves. Note that the natural join will use the attributes in common, which is x in Plan 1 and y in Plan 2. We also need to track the number of y values in the result using the assumption of preservation of values.

Plan 01 We begin...

$$T(R \bowtie S) = \frac{T(R) * T(S)}{max(V(R,x), V(S,x))} = \frac{10000 * 500}{max(1000, 200)} = 5000.$$
$$V(R \bowtie S, y) = V(R, y) = 10.$$

Next, we compute the estimated size of the second join:

$$T((R \bowtie S) \bowtie T) = \frac{T(R \bowtie S) * T(T)}{max(V(R \bowtie S, y), V(T, y))} = \frac{5000 * 2000}{max(10, 100)} = 100000.$$

Thus, the total number of tuples emitted with this logical query plan is 5000+100000 = 105000.

(We have no information about block size nor tuple size here; so, we will have to assume each tuple corresponds to a single I/O, which probably suffices for comparison at this level of detail, anyway.)

Plan 02 We begin again near the leaves, but this time joining R.y and T.y:

$$T(R \bowtie T) = \frac{T(R) * T(T)}{max(V(R,y), V(T,y))} = \frac{10000 * 2000}{max(10, 100)} = 200000.$$
$$V(R \bowtie T, x) = V(R, x) = 1000.$$

Next, we compute the estimated size of the second join:

$$T((R \bowtie T) \bowtie S) = \frac{T(R \bowtie T) * T(S)}{max(V(R \bowtie T, x), V(S, x))} = \frac{200000 * 500}{max(1000, 200)} = 100000.$$

Thus, the total number of tuples emitted with this logical query plan is 200000+100000 = 300000.

Comparison Clearly, Plan 1 is superior, as it emits just over 1/3 as many tuples as Plan 2. We note a couple other observations:

- The final operator had the same estimated cost in both plans. We would expect this, because two equivalent plans should produce the same result. This is one way to check one's work.
- The first join determined the difference between the plans. As a rule of thumb, we would want to minimise result size as much as possible as early as possible.

Question 6

Plan 01 The first operator does not change the number of tuples, due to bag semantics. Therefore:

$$\begin{aligned} T(\pi_x(R)) &= T(R) = 10000 \\ V(\pi_x(R), x) &= V(R, x) = 1000. \end{aligned}$$

The second operator removes duplicates, reducing the size to the number of unique values. Therefore:

$$\begin{aligned} T(\delta(\pi_x(R))) &= V(\pi_x(R), x) \\ &= 1000. \end{aligned}$$

The final operator will be common to both plans; so, we calculate it only to check our work, not for the sake of comparison:

$$\begin{aligned} T(Plan01) &= T(\delta(\pi_x(R))) * (1 - (1 - \frac{1}{V(R,x)})^3) \\ &= 1000 * 0.002997001 \\ &= 3. \end{aligned}$$

In total, the intermediate cost is 10000+1000 = 11000 tuples.

Plan 02 The first operator involves a disjunction of three equality predicates:

$$\begin{aligned} T(\sigma(R)) &= T(R) * (1 - (1 - \frac{1}{V(R,x)})^3) \\ &= 10000 * 0.002997001 \\ &= 30. \end{aligned}$$

The remaining number of values is given from the equality predicate. We pessimistically assume all values were matched at least once.

$$V(\sigma(R), x) = 3.$$

The second operator involves a projection, which, as before, reduces neither the number of tuples nor the number of distinct values. Thus, the second operator also emits 30 tuples.

Finally, applying the duplicate elimination operator leaves us with 3 tuples, as in Plan 1.

In total, the intermediate cost of Plan 2 is 30 + 30 = 60.

Comparison Clearly, Plan 2 is superior as it emits 0.5% of the intermediate tuples emitted by Plan 1. This is consistent with the rule of thumb that we should usually push selections as far down the query plan as possible.