CSE 344 – Final Review

August 17, 2017

Non-Parallel Query Evaluation

Consider the following schema and database instance:

```
Product(pid, name, category)
        – 10,000 tuples and 1,000 blocks
        – 40 different categories
        – Primary key (pid)
Order(store, pid, price, quantity)
        – 1,000,000 tuples and 50,000 blocks
        – prices range from $1 to $100
        – Primary key (store, pid)
```

Example query:

Compute the total revenue, for each store, from electronics costing more than $5 each:

```
SELECT o.store, sum(o.price * o.quantity)
FROM Order o, Product p
WHERE o.pid = p.pid AND o.price > 5 AND
    p.category = 'electronics'
GROUP BY o.store
```

1. Give an RA expression that:
   - computes the result of the query
   - **does not** benefit from the index on Product(pid)

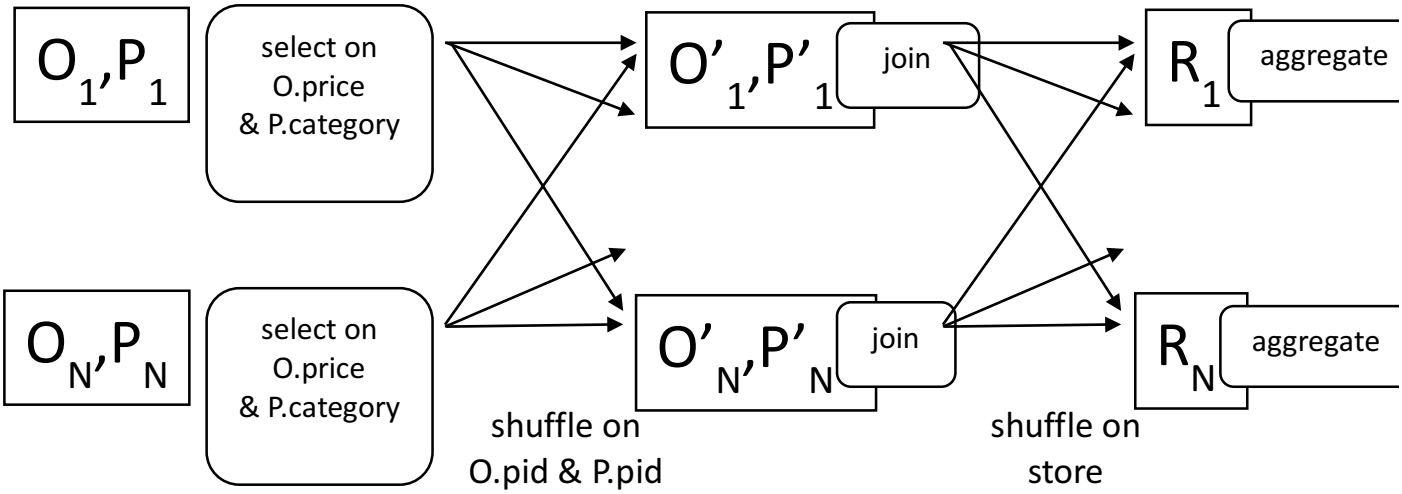
2. Estimate the cost in disk reads/writes of the RA expression from Problem 1 after filling in physical implementation details
   - assume grouping / aggregation can be done on the fly
   - use a temporary table to speed up the join where possible T1

3. Give an RA expression that:
    - computes the result of the query
    - **does** benefit from the index on Product(pid)

4. Estimate the cost in disk reads/writes of the RA expression from Problem 3 after filling in physical implementation details
    - assume grouping / aggregation can be done on the fly

Parallel Query Evaluation

The following distributed pipeline computes the result of the previous query on *N* nodes. The rows of the Product (P) and Order(O) tables are evenly distributed across the nodes.



5. Estimate the cost of executing the above pipeline
   (Assume that once read from disk, the data fits in to main memory of the nodes.)

6. Does your analysis predict a linear speedup as more nodes are added?

7. Does your analysis predict a linear scaleup as more nodes are added?

8. Describe how the cost might change if we ran a similar query with MapReduce.