

MAPREDUCE

Algorithm Design

Zareen Alamgir

Content obtained from many sources notably Jimmy Lin, Jeff Ullman, Jerome Simeon, Juliana Freire notes

Case Study

Develop an efficient Map reduce algorithm to solve the following real-world problem.

- Find set of common friends between any two users on Facebook.



Case Study

- Facebook has a list of friends
 - ⇒ Note that friends are a bi-directional on Facebook.
 - ⇒ If I'm your friend, you're mine.



- FB have lots of disk space and they serve hundreds of millions of requests every day.
- They pre-compute calculations when they can to reduce the processing time of requests.



Case Study

- **One common processing request is the ‘Common Friends’ that is you visit someone's profile, you see a list of friends that you have in common.**
- **This list doesn't change frequently so it'd be wasteful to recalculate it every time you visited the profile.**
- **Your task is to use map reduce and calculate everyone's common friends once a day and store those results. Later on it's just a quick lookup.**



Case Study

- Assume the friends are stored as **Person->[List of Friends]**
 - A -> B C D
 - B -> A C D E
 - C -> A B D E
 - D -> A B C E
 - E -> B C D
-
- Give pseudo-code for Map and Reduce functions for the above task.

Case Study

- Assume the friends are stored as **Person->[List of Friends]**
- A -> B C D
- B -> A C D E
- C -> A B D E
- D -> A B C E
- E -> B C D
- BASIC IDEA
 - A -> B C D
 - Pair approach vs stripe approach

Basic set operations

Primitives

- Projection (π)
- Selection (σ)
- Cartesian product (\times)
- Set union (\cup)
- Set difference ($-$)

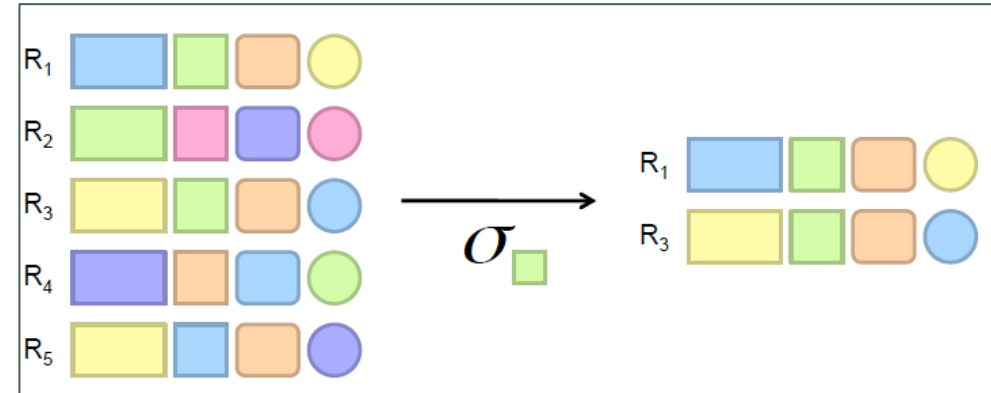
Other operations

- Group by... aggregation
- ...

Relational-Algebra Operations

Selection:

- Select only those tuples of relation R that satisfy condition C.



Map Function:

- For each record (tuple t) in R, test if it satisfies C.
- Emit only the tuple that satisfy t

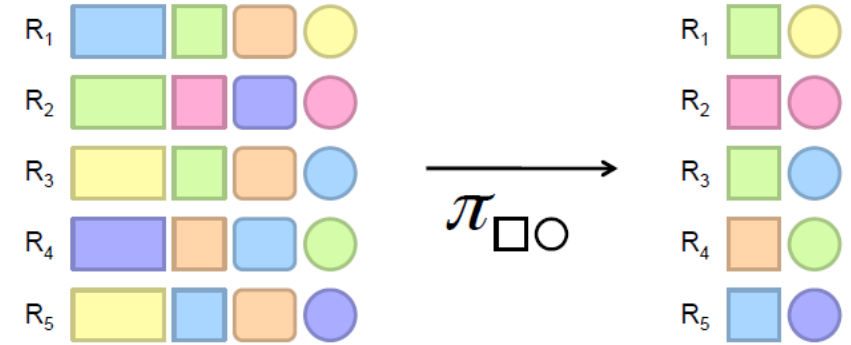
Reduce Function:

- No Reducer Necessary (unless you want to do something else)

Relational-Algebra Operations

Projection:

- For each tuple produce only the subset of attributes as specified.



Map Function:

- process each tuple, re-emit with only projected attributes
- Can be pipelined with selection

Reduce Function:

- No reducers necessary (unless to do something else)
- You can use reducer to output only unique values.
 - For each key t produced by any of the Map tasks, if there are more than one values than the Reduce function produces exactly one output t.

Relational-Algebra Operations

Union, Intersection, and Difference:

- These well-known set operations apply to the sets of tuples in two relations that have the same schema.

UNION

- Map Function: Turn each input tuple t into a key-value pair (t, ϕ) .
- Reduce Function: Associated with each key t there will be either one or two values. Produce output t in either case.

Relational-Algebra Operations

Grouping and Aggregation:

- Aggregation functions :AVG, MAX, MIN, SUM, COUNT, ...
- Map Function:
 - Map over dataset, emit tuples, keyed by group by attribute
- Framework automatically groups values by group by attribute
- Reduce Function:
 - Compute aggregation function in reducer
- Optimize with combiners, in-mapper combining