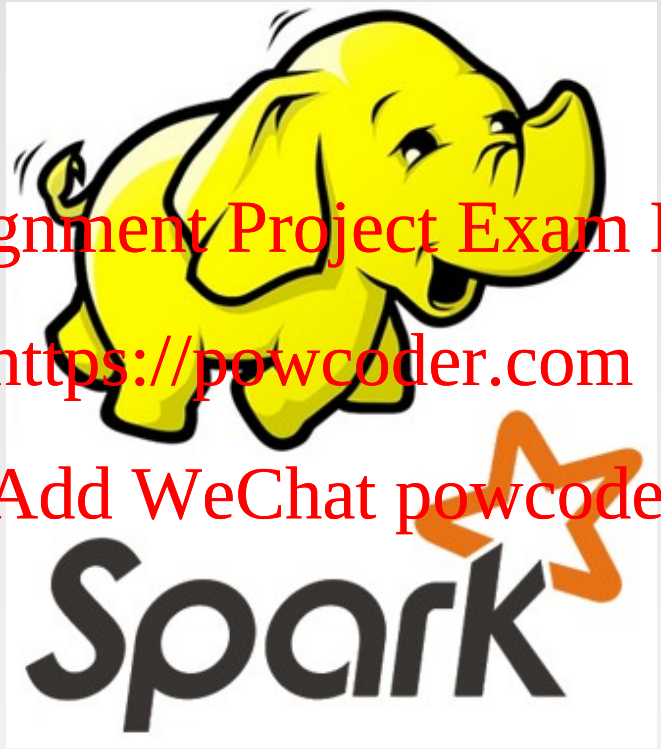
COMP9313: Big Data Management

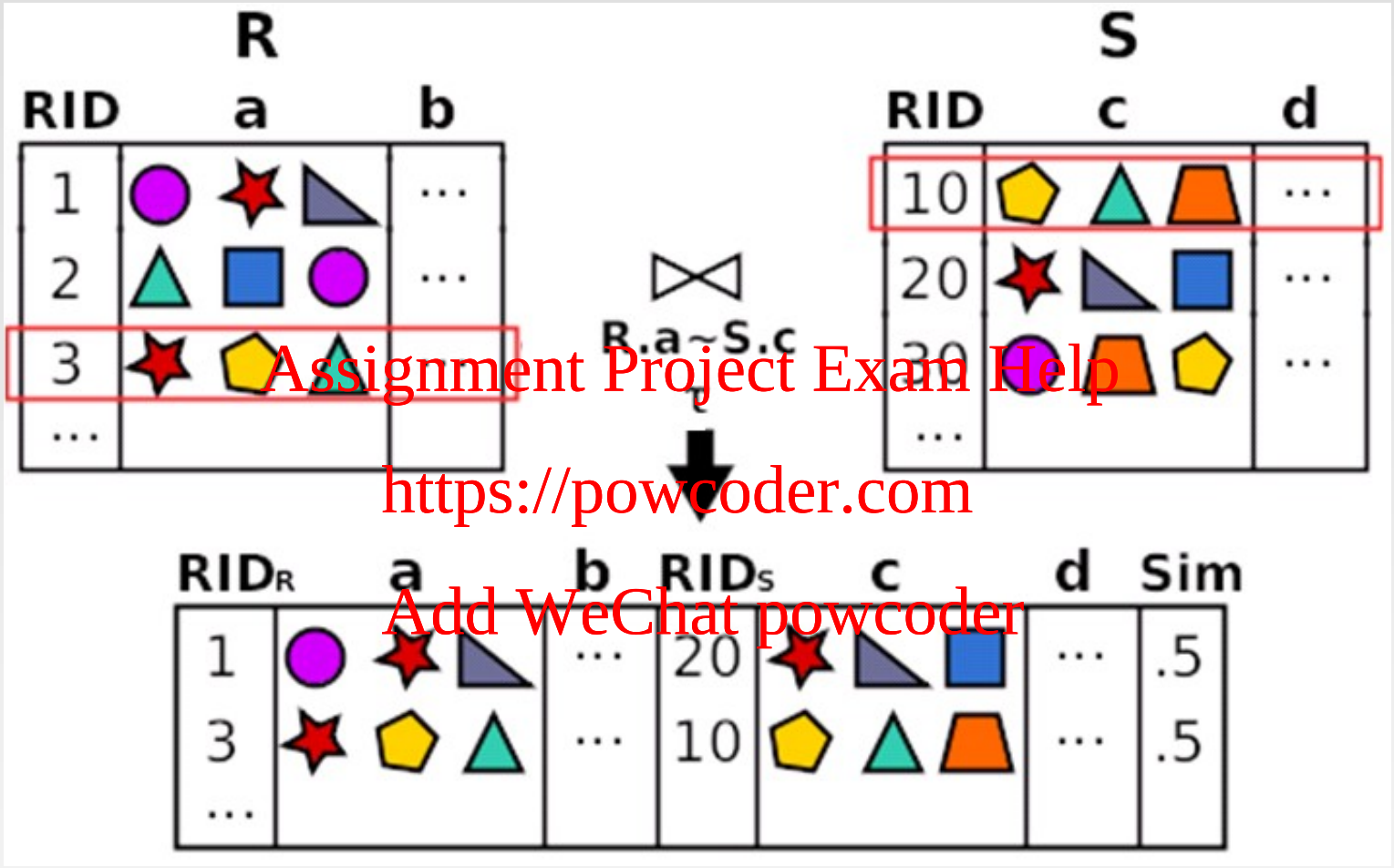


Lecturer: Xin Cao

Course web site: http://www.cse.unsw.edu.au/~cs9313/

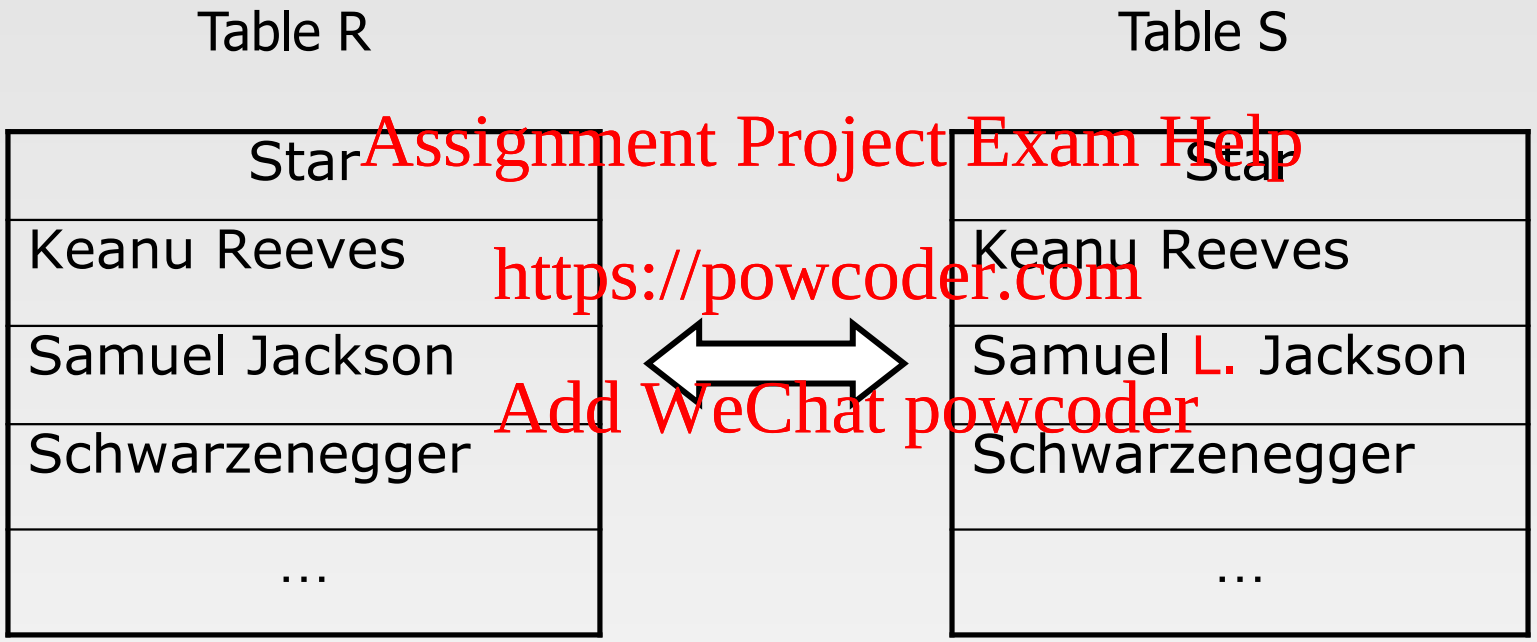
Set Similarity Join on Hadoop

Set-Similarity Join

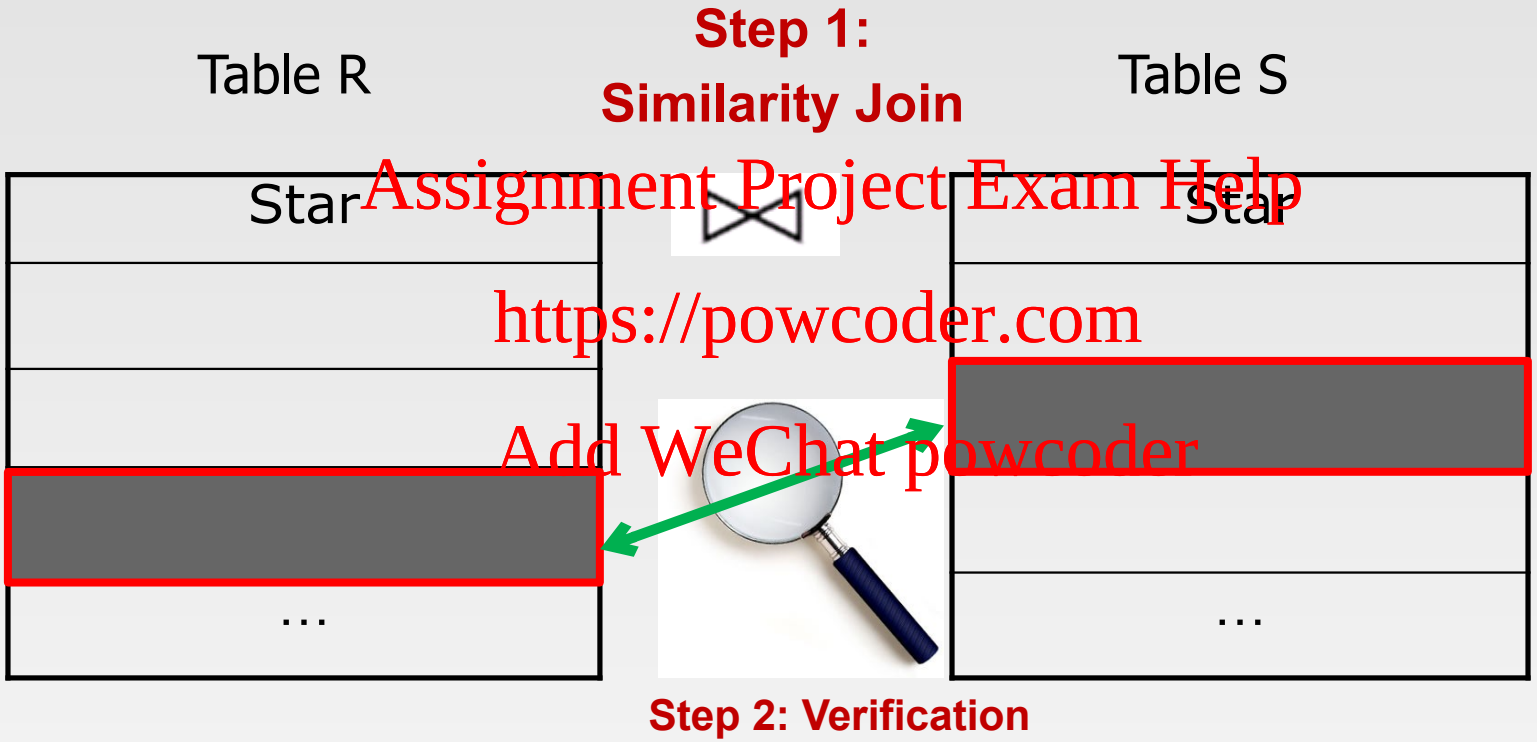


Finding pairs of records with a similarity on their join attributes > t

Application: Record linkage



Two-step Solution



Why Hadoop?

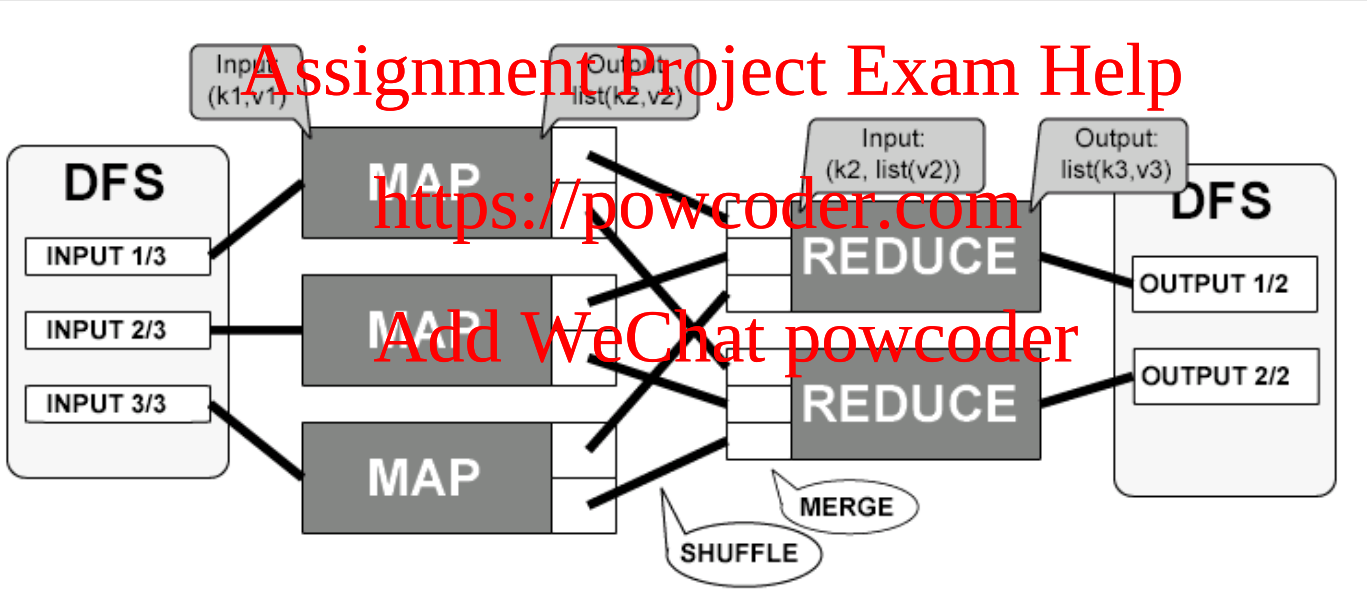
- Large amounts of data
- Data or processing does not fit in one machine

- Assumptions:
  - Self join: S=S
  - Two similar sets share at least 1 token

- Efficient Parallel Set-Similarity Joins Using Hadoop (SIGMOD'10)

A naïve solution

- Map: <23, (a,b,c)> → (a, 23), (b, 23), (c, 23)
- Reduce:(a,23),(a,29),(a,50), …→ Verify each pair (23, 29), (23, 50), (29, 50) … …

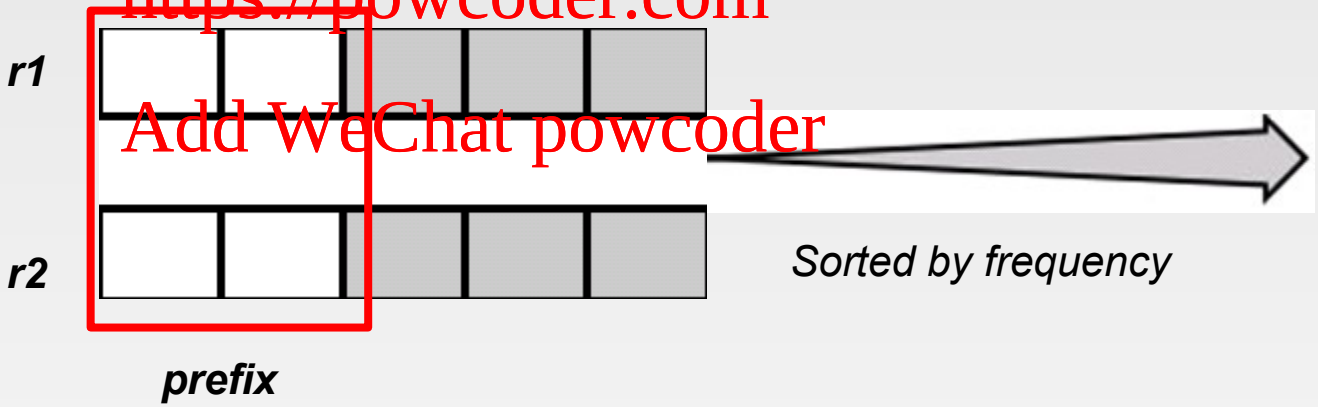


- Too much data to transfer ☹
- Too many pairs to verify ☹.

Solving frequency skew: prefix filtering

- Sort tokens by frequency (ascending)

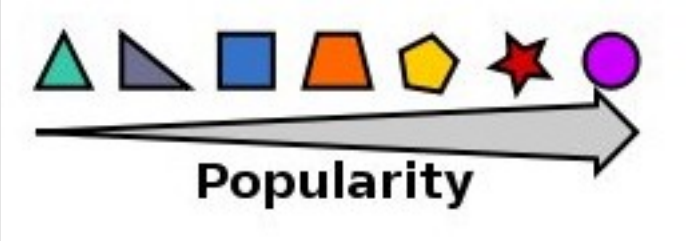
- Prefix of a set: least frequent tokens



- Prefixes of similar sets should share tokens

Chaudhuri, Ganti, Kaushik: A Primitive Operator for Similarity Joins in Data Cleaning. ICDE'06

Prefix filtering: example



Record 1

Record 2

- Each set has 5 tokens
- “Similar”: they share at least 4 tokens
- Prefix length: 2

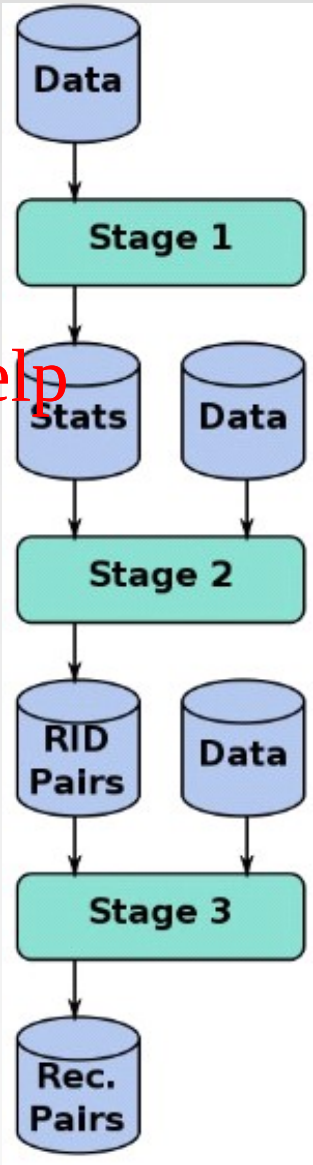
Hadoop Solution: Overview

- Stage 1: Order tokens by frequency

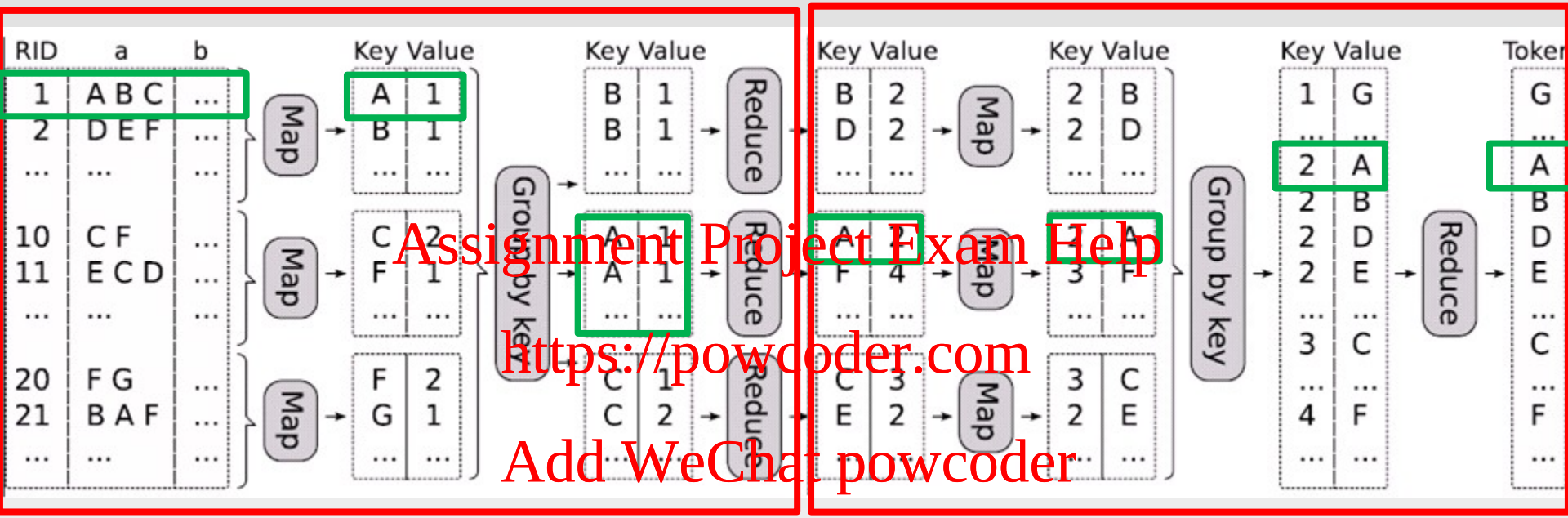
(Already done in the given example data)

- Stage 2: Finding "similar" id pairs (verification)

- Stage 3: remove duplicates



Stage 1: Sort tokens by frequency



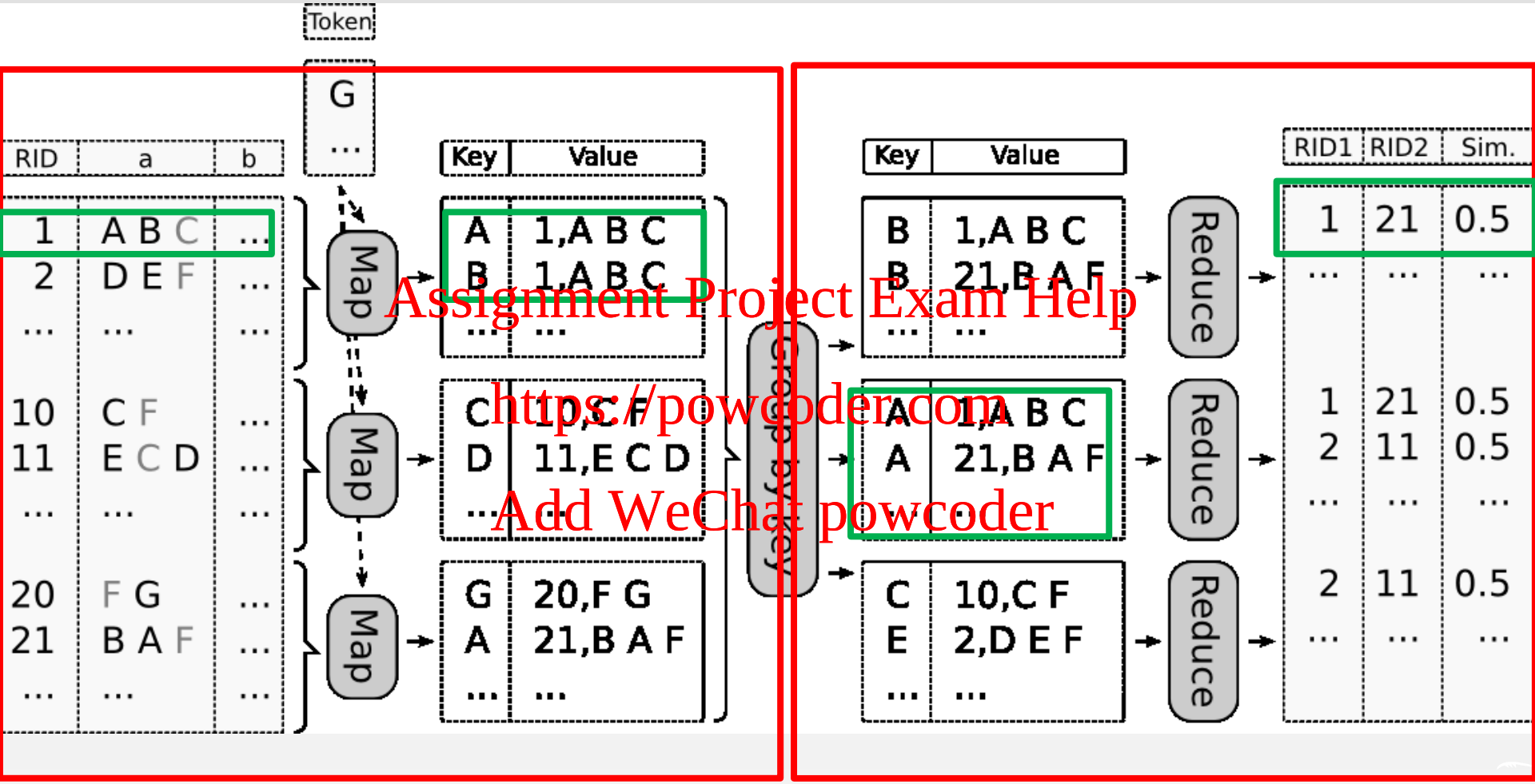
Compute token frequencies

Sort them

MapReduce phase 1

MapReduce phase 2

Stage 2: Find “similar” id pairs



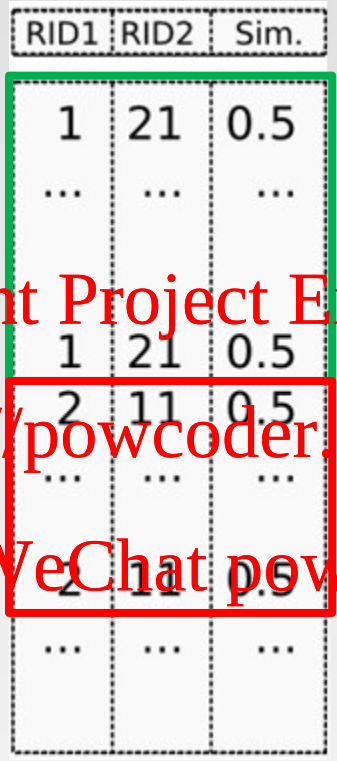
Partition using prefixes

Verify similarity

Compute the Length of Shared Tokens

- Jaccard Similarity: **sim(r, s) = |r∩s|/|r∪s|**
- If **sim(r, s) >= τ, l = |r∩s| >= |r∪s| * τ >= max(|r|, |s|) * τ**
- Given a record r, you can compute the prefix length as p = |r| - l + 1
- r and s is a candidate pair, they must share at least one token in the first (|r| - l + 1) tokens
- Given a record r = (A, B, C, D) and p = 2, the mapper emits (A, r) and (B, r)

Stage 3: Remove Duplicates



RID1	RID2	Sim.
1	21	0.5
...	...	...
1	21	0.5
2	11	0.5
...	...	...
2	11	0.5
...	...	...

More Optimization Strategies

- **It is your job!!!**

- **The faster the better**