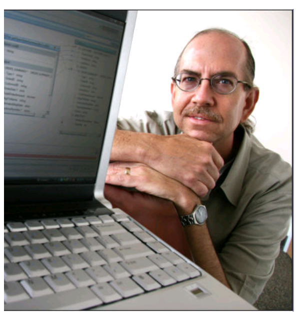
Efficient Parallel Set-Similarity Joins Using Hadoop

Chen Li

UCIrvine
University of California, Irvine



Joint work with
Michael Carey and Rares Vernica

YAHOO! PRESENTS

hadoop SUMMIT 2010

Motivation: Data Cleaning

Find movies starring Tom Hanks

Star	Title	Year	Genre
Keanu Reeves	The Matrix	1999	Sci-Fi
Tom Hanks	Toy Story 3	2010	Animation
Schwarzenegger	The Terminator	1984	Sci-Fi
Samuel Jackson	The man	2006	Crime

Movies starring S..warz…ne…ger?

Star	Title	Year	Genre
Keanu Reeves	The Matrix	1999	Sci-Fi
Tom Hanks	Toy Story 3	2010	Animation
Schwarzenegger	The Terminator	1984	Sci-Fi
Samuel Jackson	The man	2006	Crime

Similarity Search

Find movies with a star "similar to" Schwarrzenger.

Star	Title	Year	Genre
Keanu Reeves	The Matrix	1999	Sci-Fi
Samuel Jackson	Iron man	2008	Sci-Fi
Schwarzenegger	The Terminator	1984	Sci-Fi
Samuel Jackson	The man	2006	Crime

Record linkage

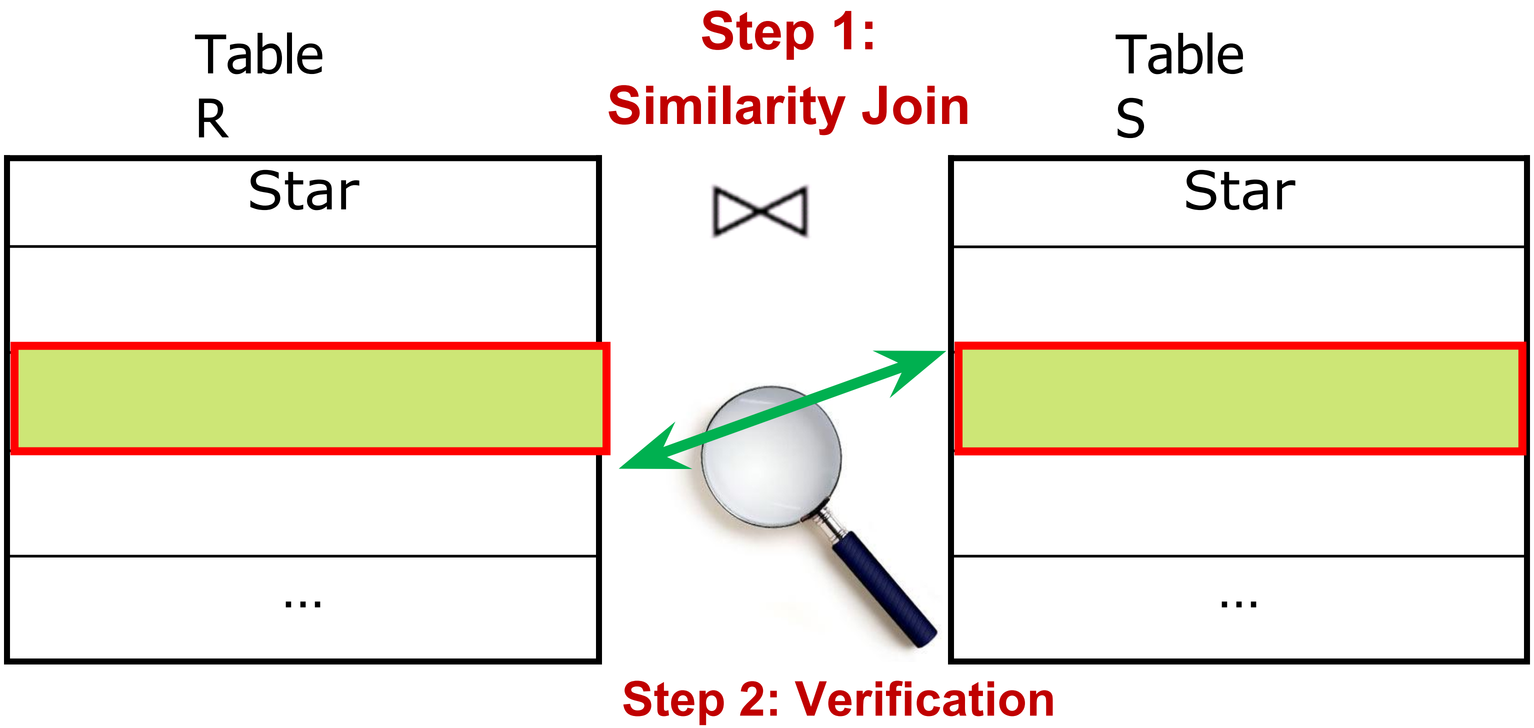
Table R

Star
Keanu Reeves
Samuel Jackson
Schwarzenegger
...

Table S

Star
Keanu Reeves
Samuel L. Jackson
Schwarzenegger
...

Two-step solution



Focus of this talk

- Similarity join for large data sets
- Techniques applicable to other domains, e.g.:
  - Finding similar documents
  - Finding customers with similar patterns

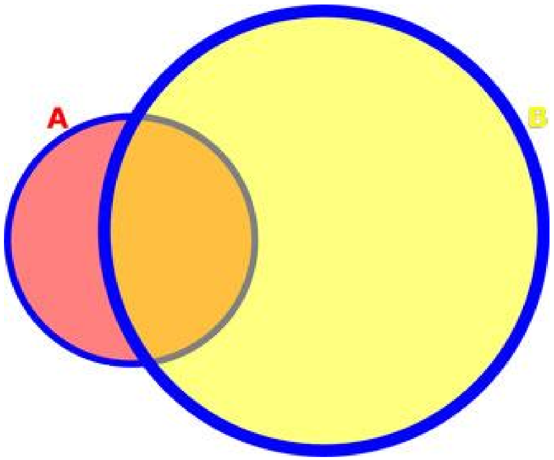
Talk Outline

- Formulation: set-similarity join
- Hadoop-based solutions
- Experiments

Set-similarity functions

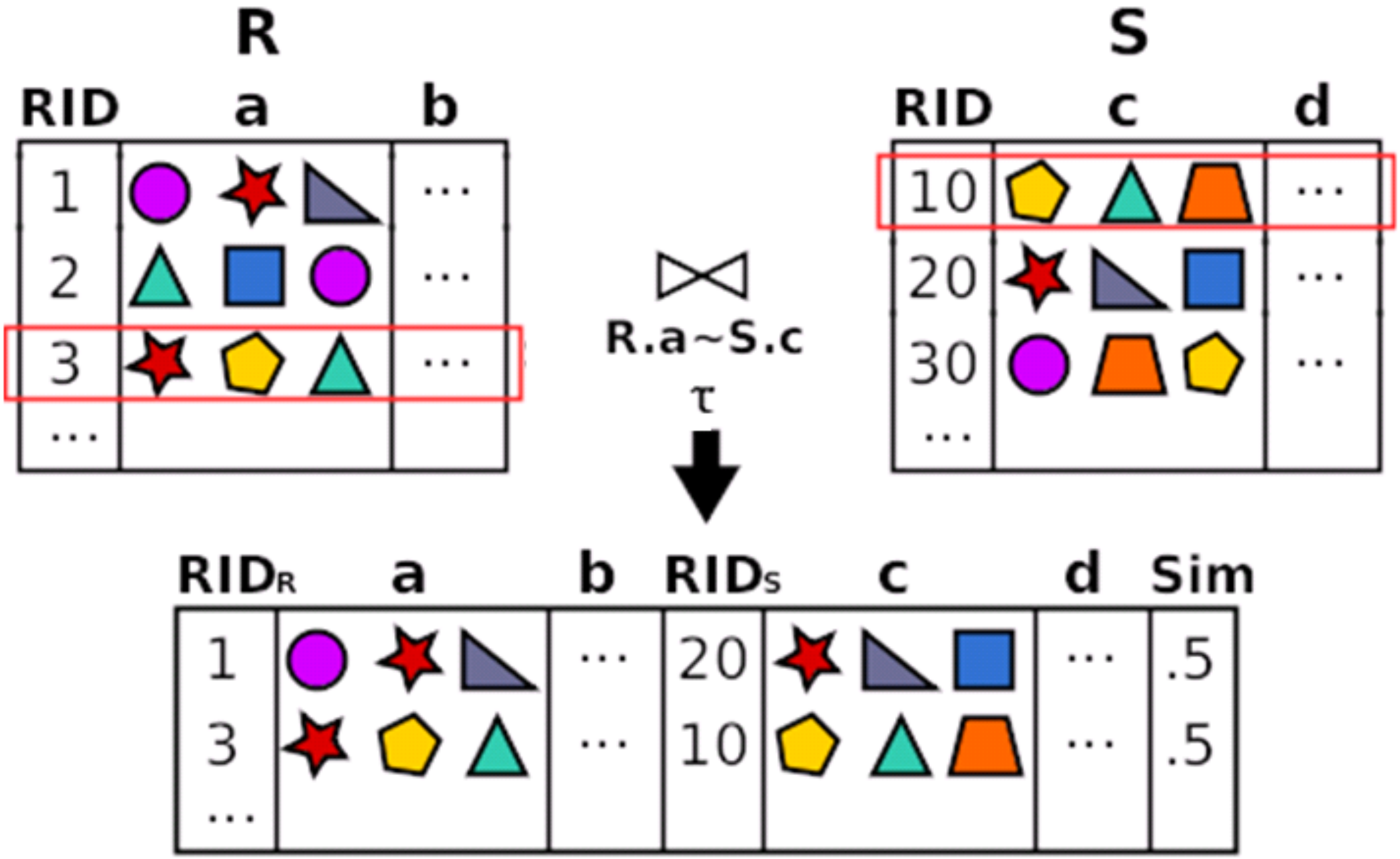
- Jaccard
- Dice
- Cosine
- Hamming
- …

$$Jaccard(A,B) = \frac{|A \cap B|}{|A \cup B|} \geq t$$



All solvable in this framework

Set-Similarity Join



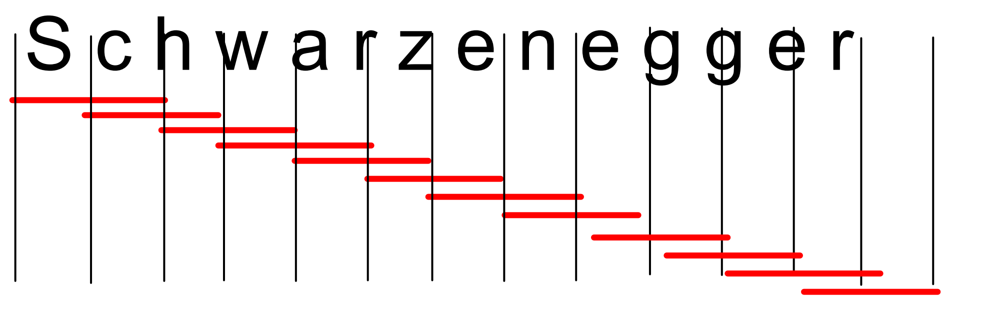
Finding pairs of records with a similarity on their join attributes > t

Why this formulation?

- Word tokens:

“Samuel L. Jackson” → {Samuel, L., Jackson}
“Samuel Jackson” → {Samuel, Jackson}

- Gram tokens:



Talk Outline

- Formulation of set-similarity join
- → Hadoop-based solutions
- Experiments

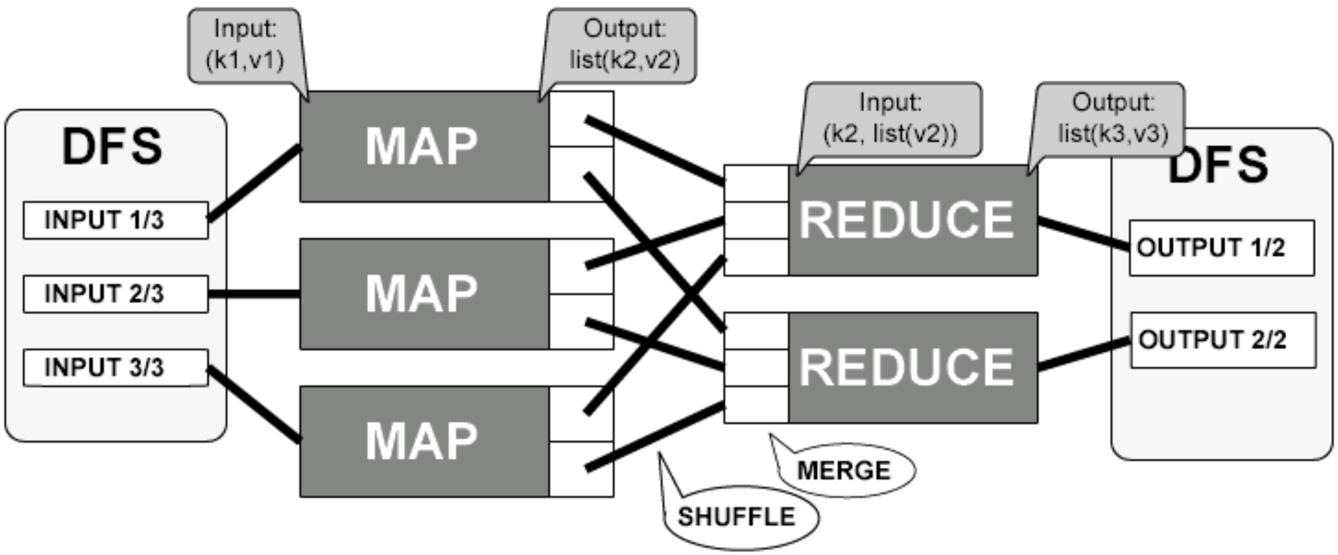
Why Hadoop?

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

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

A naïve solution

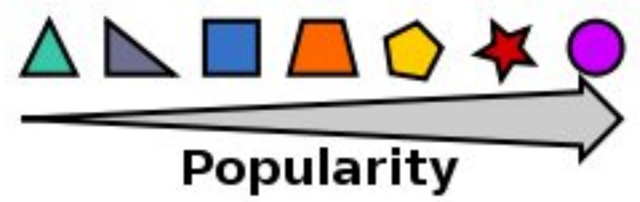
- Map: <23, (a,b,c)> → (a, 23), (b, 23), (c, 23)
- Reduce:(a,23),(a,29),(a,50), …→ Verify each pair



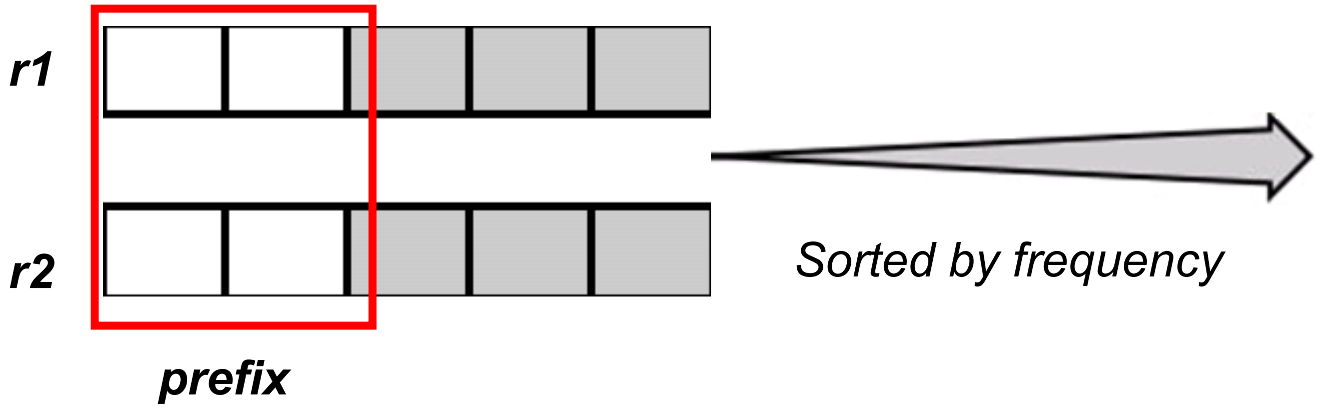
- Too much data to transfer ☹
- Too many pairs to verify ☹.

Solving frequency skew: prefix filtering

- Sort tokens by frequency (ascending)



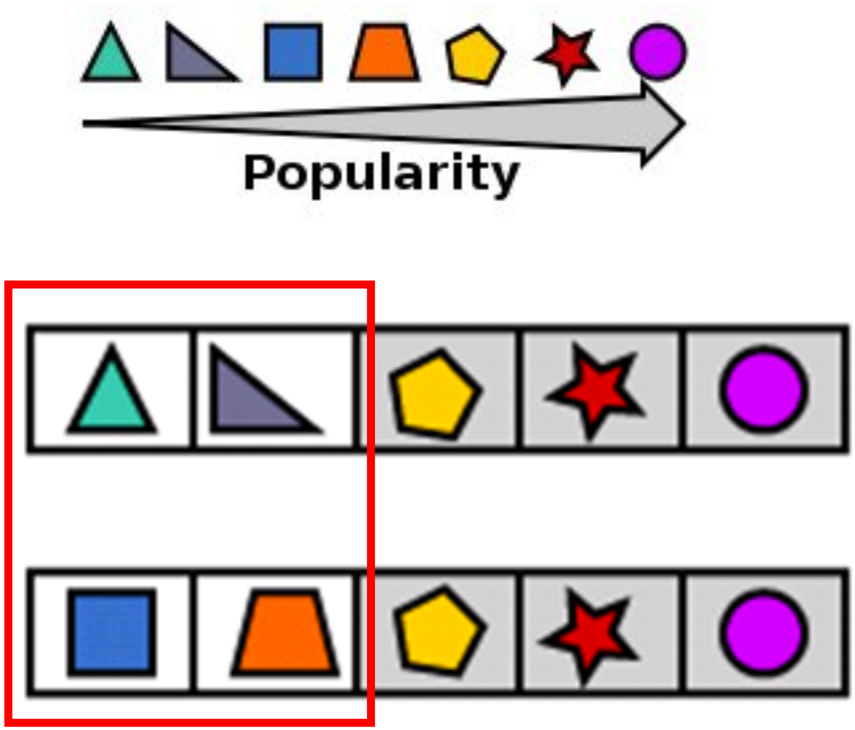
- Prefix of a set: least frequent tokens



- Prefixes of similar sets should share tokens

Chaudhuri, Ganti, Kaushik: A Primitive Operator for Similarity Joins in Data Cleaning. ICDE 2006: 5

Prefix filtering: example



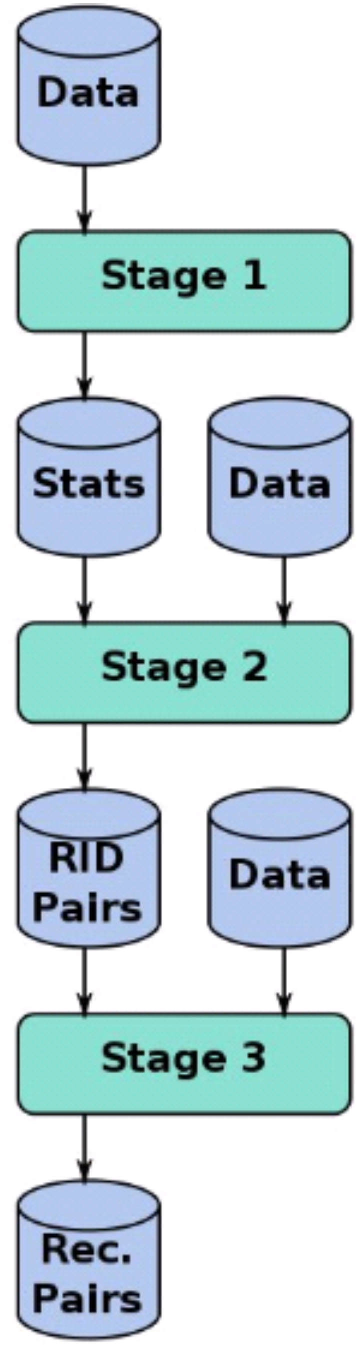
Record 1

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
- Stage 2: Finding “similar” id pairs
- Stage 3: id pairs → record paris

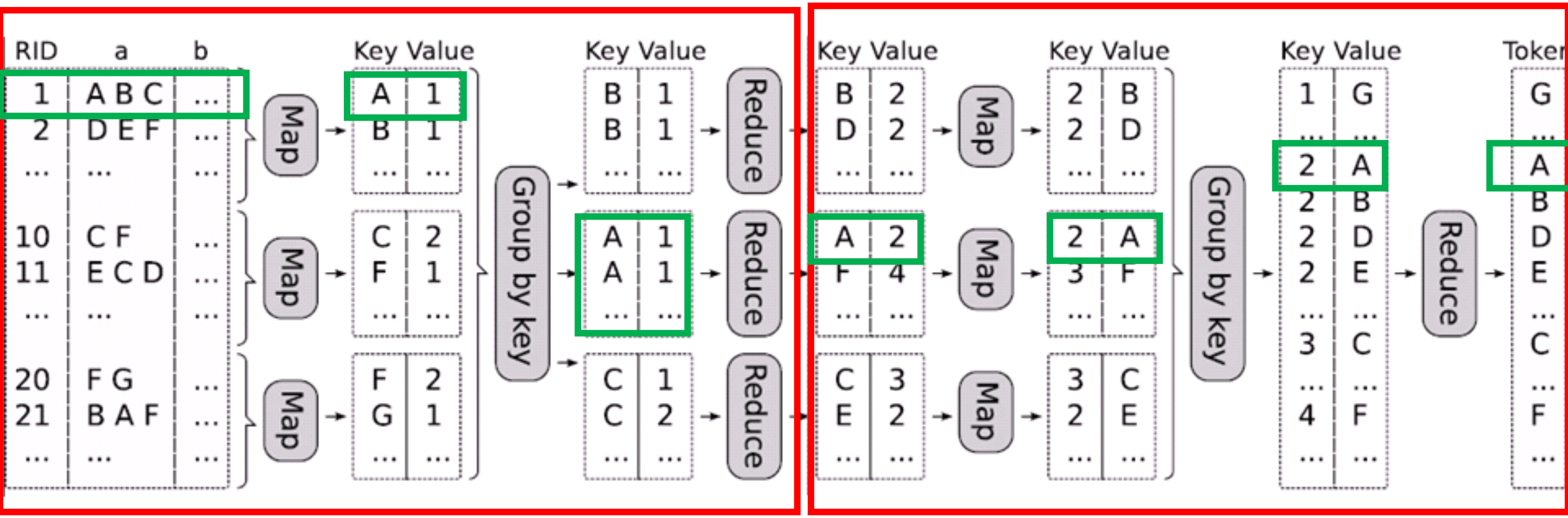


*

NTS

UMMIT 2010

Stage 1: Sort tokens by frequency



Compute token frequencies

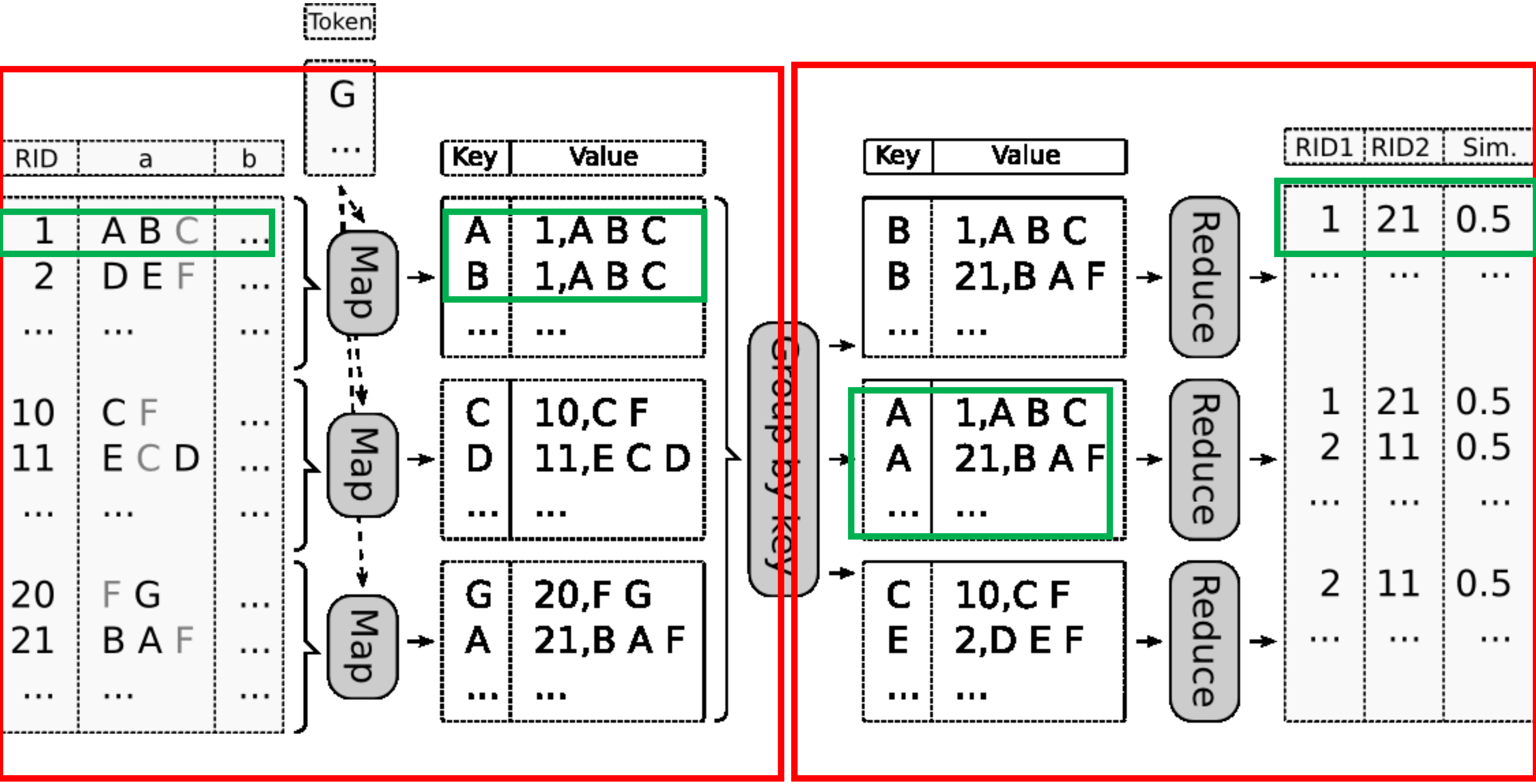
Sort them

MapReduce phase 1

MapReduce phase 2

*

Stage 2: Find "similar" id pairs

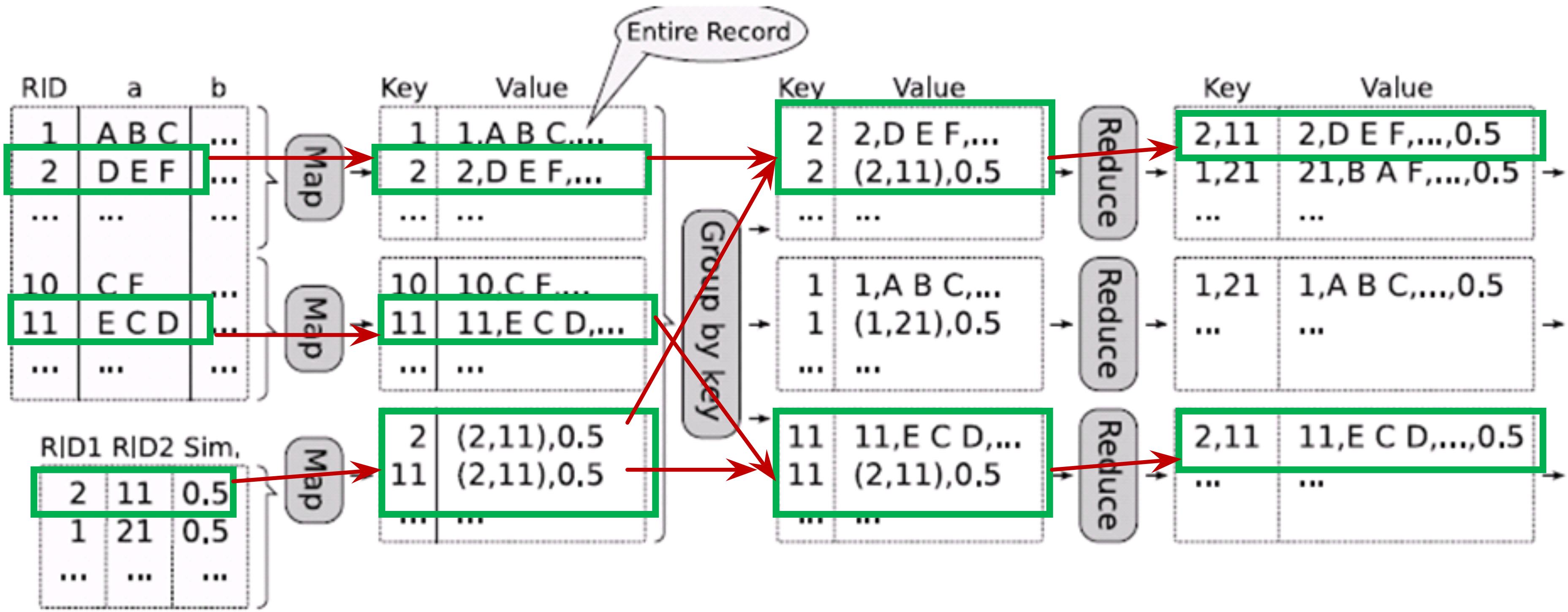


Partition using prefixes

Verify similarity

*

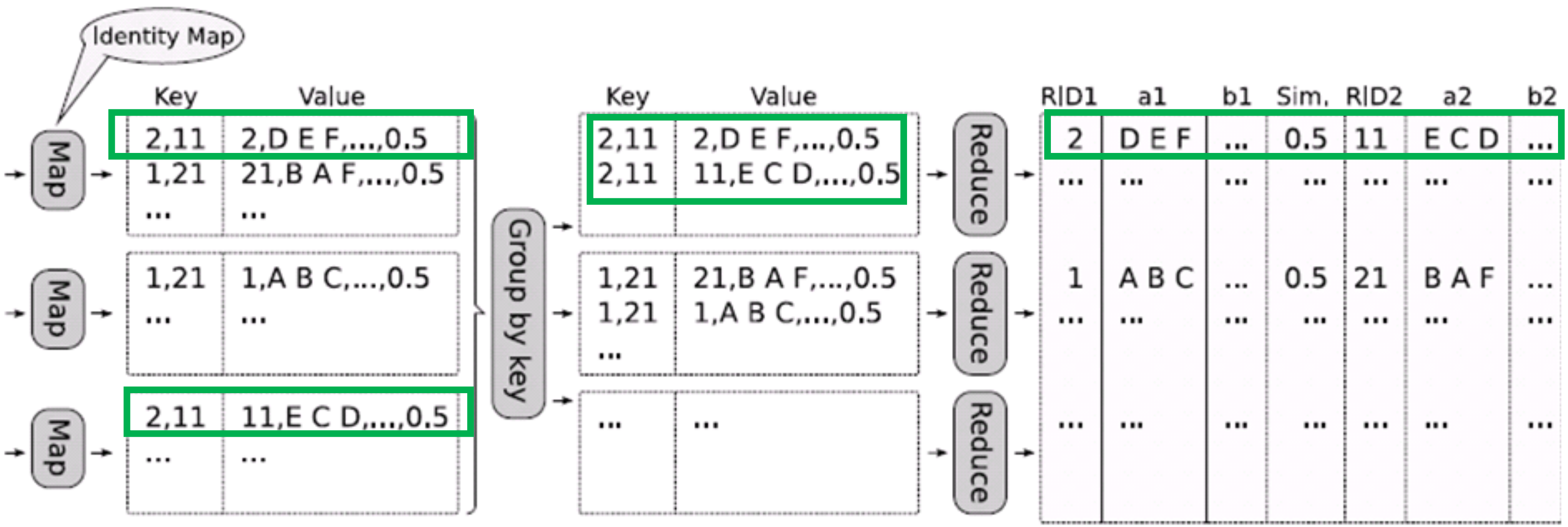
Stage 3: id pairs → record pairs (phase 1)



Bring records for each id in each pair

*

Stage 3: id pairs → record pairs (phase 2)



Join two half filled records

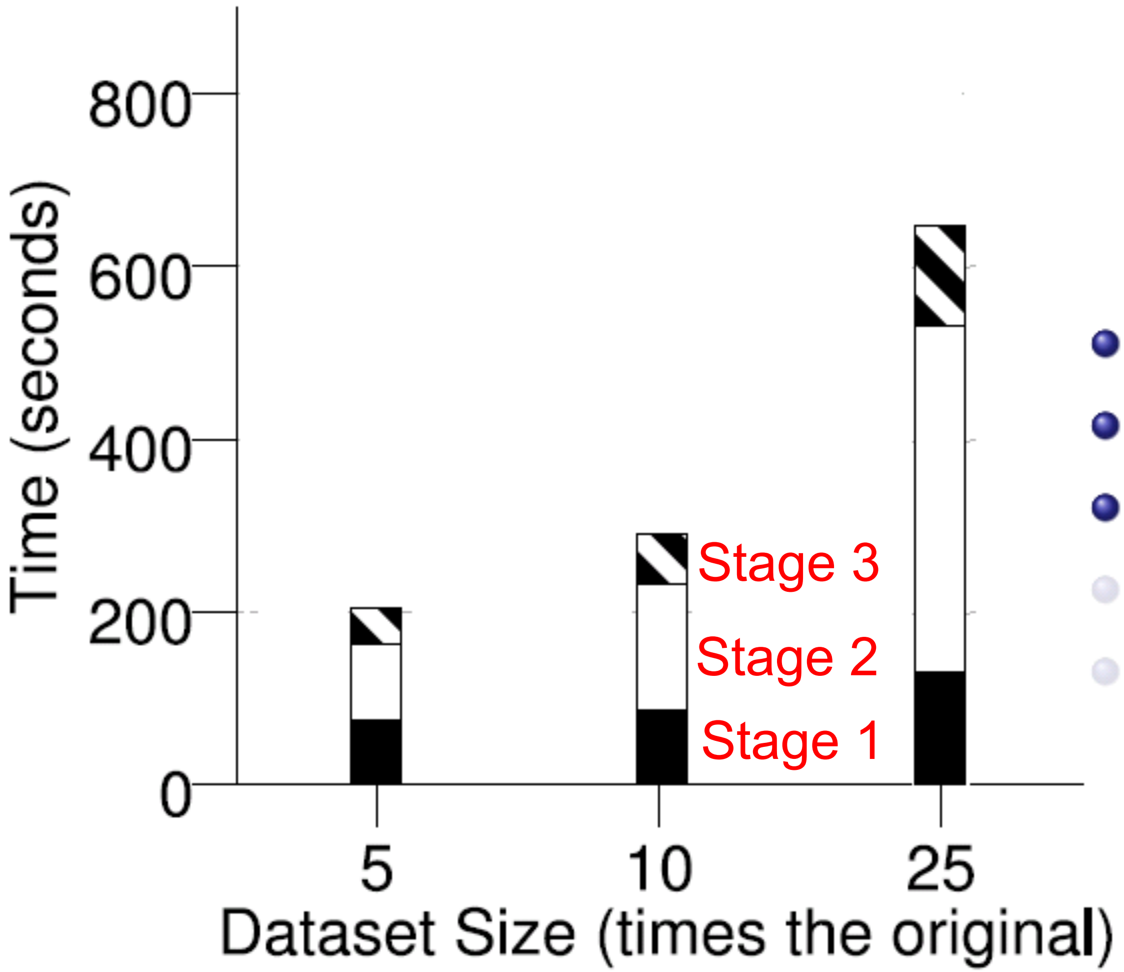
Talk Outline

- Formulation of set-similarity join
- Hadoop-based solutions
- → Experiments

Experimental Setting

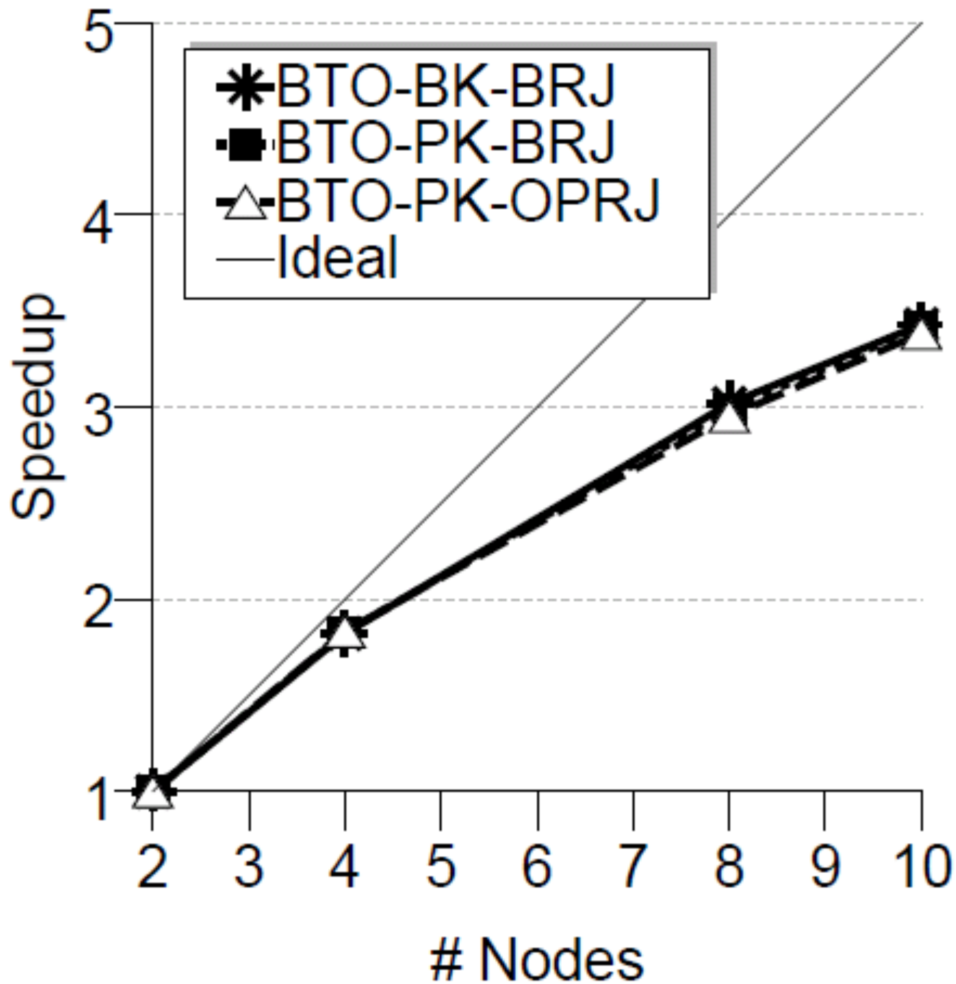
- Hardware
  - 10-node IBM x3650 cluster
  - Intel Xeon processor E5520 2.26GHz with four cores
  - Four 300GB hard disks
  - 12GB RAM
- Software
  - Ubuntu 9.06, 64-bit, server edition OS
  - Java 1.6, 64-bit, server
  - Hadoop 0.20.1
- Datasets: publications (DBLP and CITESEERX)

Running time

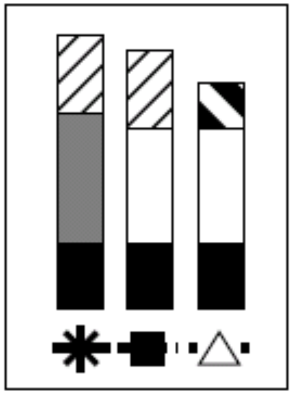


- Self-join DBLP $\times n$
- $n \in [5, 25]$
- 10-node cluster
- Best time
- Bulk of the time

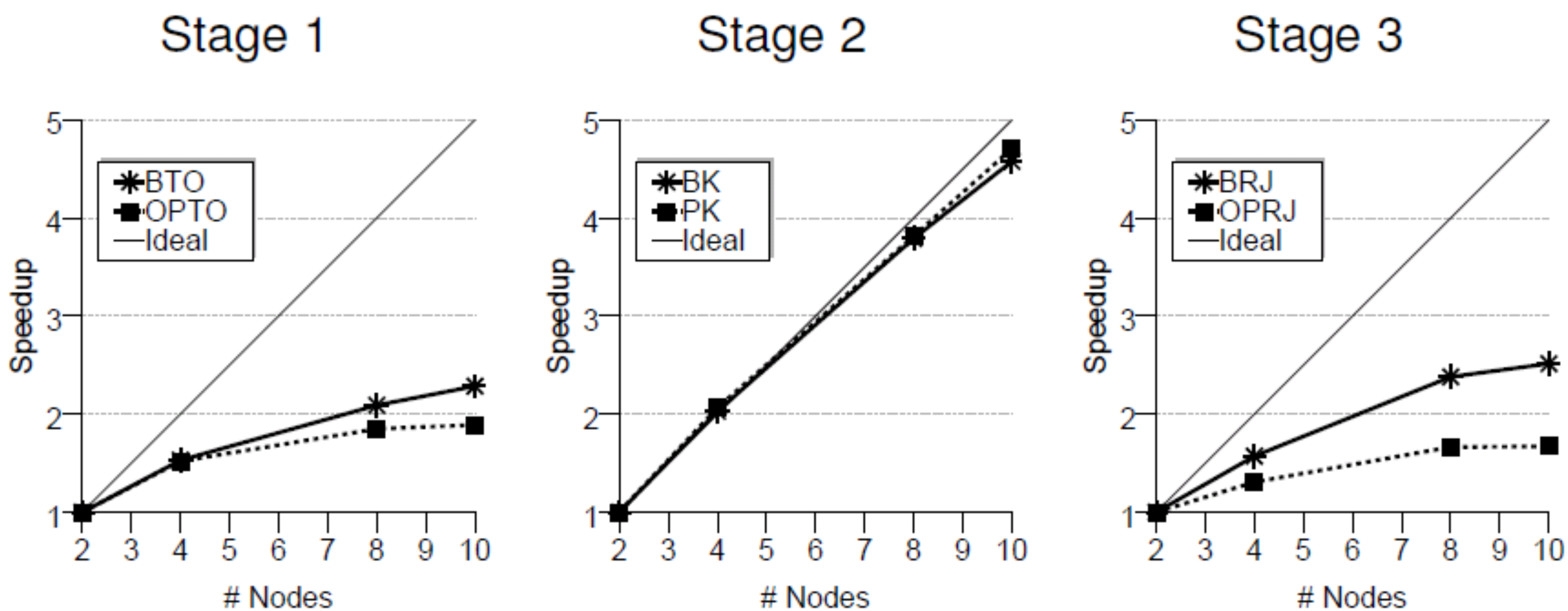
Speedup



- Relative running time
- Self-join DBLP $\times 10$
- Different cluster sizes



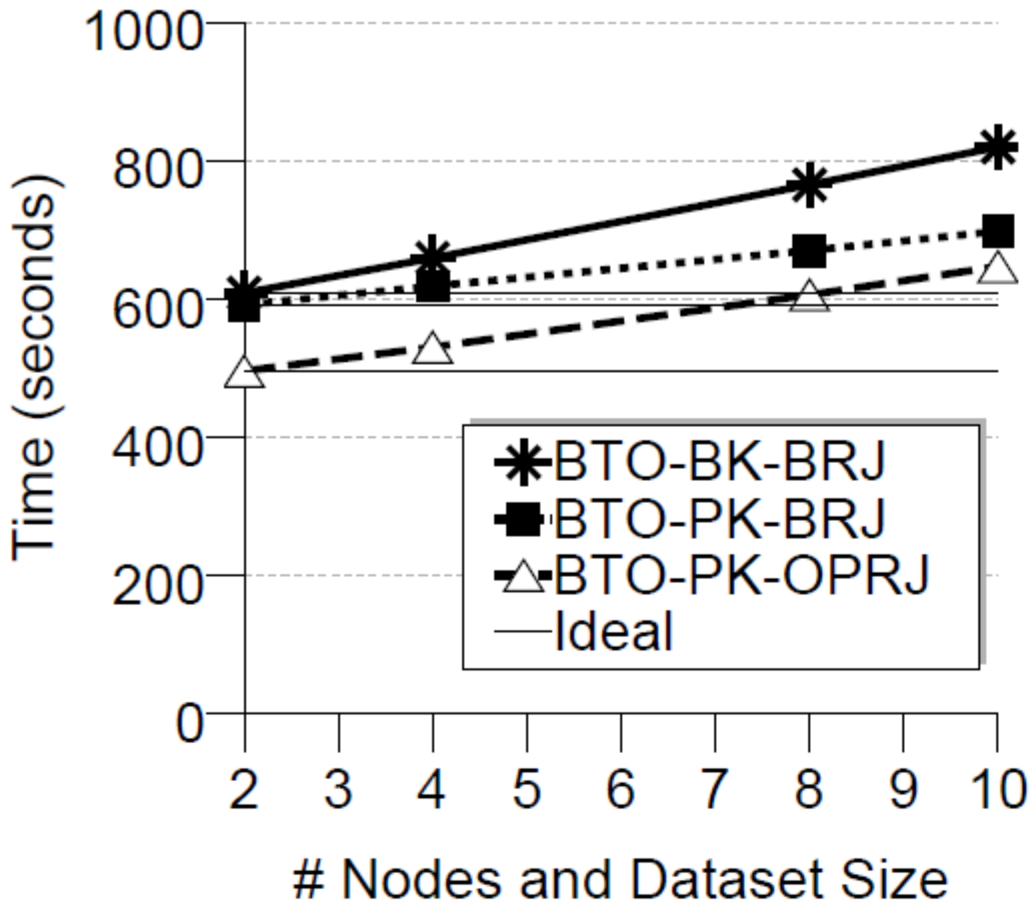
Speedup Breakdown



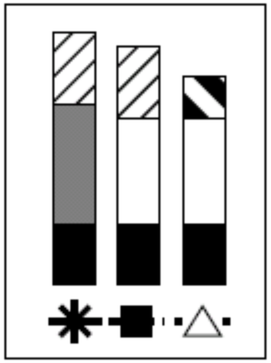
- Relative running time
- Self-join DBLP × 10
- Different cluster sizes

Stage 2 has good speedup

Scaleup



- Running time
- Self-joining DBLP $\times n$
- $n \in [5, 25]$
- Proportional cluster



Good scaleup

Thank you

Chen Li @ UC Irvine

Source code available at:

http://asterix.ics.uci.edu/fuzzyjoin-mapreduce/

Acknowledgements:
NSF, Google, IBM.

YAHOO! PRESENTS

hadoop SUMMIT 2010