Efficient parallel set-similarity joins using MapReduce

Speaker: Bibek Paudel
Tutor: Jörg Schad

January 28, 2011

Set Similarity

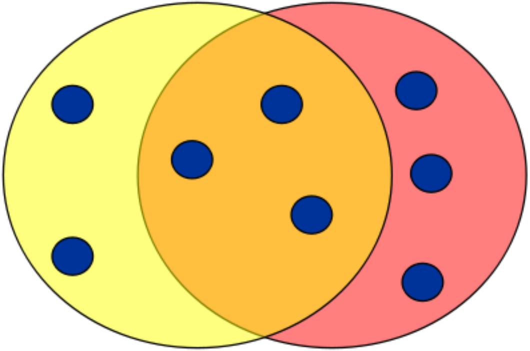


Figure: Set Similarity (Jaccard) is: 3/8

Examples and Uses

- Detect Spam

Examples and Uses

- Detect Spam
- Detect mirrored web pages

Examples and Uses

- Detect Spam
- Detect mirrored web pages
- Detect plagiarism

Examples and Uses

- Detect Spam
- Detect mirrored web pages
- Detect plagiarism
- Information Extraction

Examples and Uses

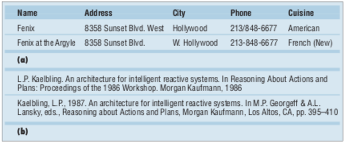
- Detect Spam
- Detect mirrored web pages
- Detect plagiarism
- Information Extraction
- Distance between strings or documents

Different Metrics

- Edit Distance
- Hamming Distance
- Overlap coefficient
- Similarity measures

Different Metrics

- Edit Distance
- Hamming Distance
- Overlap coefficient
- Similarity measures



Name	Address	City	Phone	Cuisine
Fenix	8358 Sunset Blvd. West	Hollywood	213/848-6677	American
Fenix at the Argyle	8358 Sunset Blvd.	W. Hollywood	213-848-6677	French (New)

(a)

L.P. Kaelbling. An architecture for intelligent reactive systems. In Reasoning About Actions and Plans: Proceedings of the 1986 Workshop. Morgan Kaufmann, 1986

Kaelbling, L.P., 1987. An architecture for intelligent reactive systems. In M.P. Georgeff & A.L. Lansky, eds., Reasoning about Actions and Plans, Morgan Kaufmann, Los Altos, CA, pp. 395–410

(b)

Figure: Sample duplicate records[a]

[a] Adaptive Name Matching in Information Integration, Bilenko et al, IEEE Computer Society

Challenges

- Find similarity between all pairs?
- Find exact similarity or an approximation?

Challenges

- Find similarity between all pairs?
- Find exact similarity or an approximation?

- How to reduce the number of comparisons?
- How to use filtering?

Existing Methods

- length filter

Existing Methods

- length filter
- suffix and prefix filter

Existing Methods

- length filter
- suffix and prefix filter
- PPJoin [Example on board]

How to scale it up?

- Attractions of distributed system

How to scale it up?

- Attractions of distributed system
- MapReduce?

How to scale it up?

- Attractions of distributed system
- MapReduce?
- Working of MapReduce

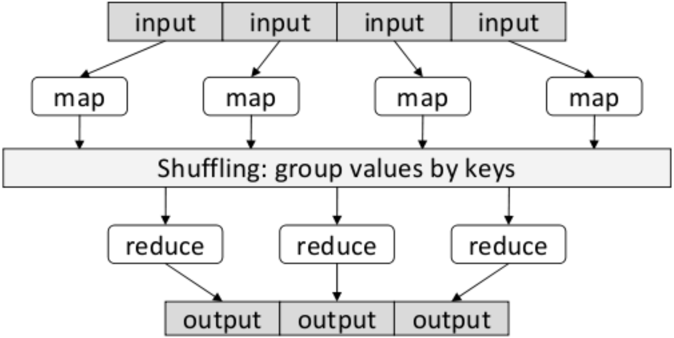


Figure: MapReduce

The algorithm of the paper

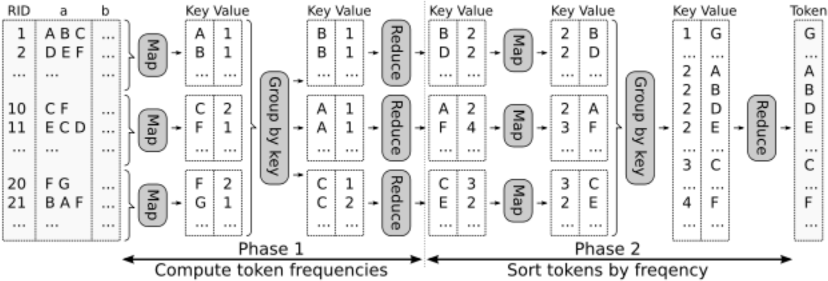
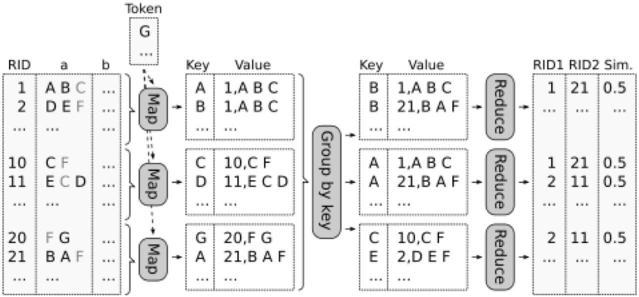


Figure: Phase 1

The algorithm of the paper



RID	a	b
1	A B C	...
2	D E F	...
...	...	...
10	C F	...
11	E C D	...
...	...	...
20	F G	...
21	B A F	...
...	...	...

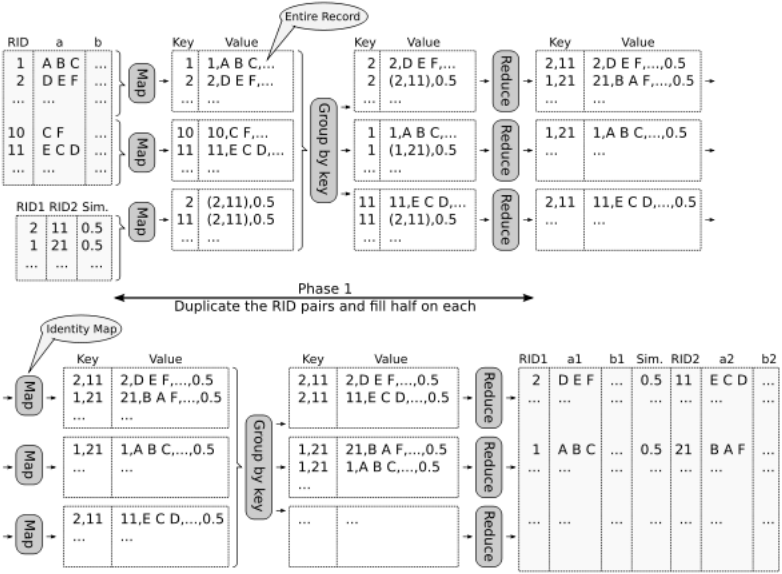
Key	Value
A	1,A B C
B	1,A B C
...	...
C	10,C F
D	11,E C D
...	...
G	20,F G
A	21,B A F
...	...

Key	Value
B	1,A B C
B	21,B A F
...	...
A	1,A B C
A	21,B A F
...	...
C	10,C F
E	2,D E F
...	...

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

Figure: Phase 2

The algorithm of the paper



RID	a	b
1	A B C	...
2	D E F	...
...	...	...
10	C F	...
11	E C D	...
...	...	...

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

Key	Value
1	1,A B C,...
2	2,D E F,...
...	...
10	10,C F,...
11	11,E C D,...
...	...
2	(2,11),0.5
11	(2,11),0.5
...	...

Key	Value
2	2,D E F,...
2	(2,11),0.5
...	...
1	1,A B C,...
1	(1,21),0.5
...	...
11	11,E C D,...
11	(2,11),0.5
...	...

Key	Value
2,11	2,D E F,...,0.5
1,21	21,B A F,...,0.5
...	...
1,21	1,A B C,...,0.5
...	...
2,11	11,E C D,...,0.5
...	...

Key	Value
2,11	2,D E F,...,0.5
1,21	21,B A F,...,0.5
...	...
1,21	1,A B C,...,0.5
...	...
2,11	11,E C D,...,0.5
...	...

Key	Value
2,11	2,D E F,...,0.5
2,11	11,E C D,...,0.5
1,21	21,B A F,...,0.5
1,21	1,A B C,...,0.5
...	
...	...

RID1	a1	b1	Sim.	RID2	a2	b2
2	D E F	...	0.5	11	E C D	...
...	...	...	...	...	...	...
1	A B C	...	0.5	21	B A F	...
...	...	...	...	...	...	...
...	...	...	...	...	...	...

Alternatives for each phase

- One phase token ordering

Alternatives for each phase

- One phase token ordering
- Kernel

Alternatives for each phase

- One phase token ordering
- Kernel
- One phase for phase 3

Alternatives for each phase

- One phase token ordering
- Kernel
- One phase for phase 3
- Total three M/R jobs

Alternatives for each phase

- One phase token ordering
- Kernel
- One phase for phase 3
- Total three M/R jobs
- R-S Join and Self-Join

Issues and shortcomings

- Dictionary size
- Candidates size

Issues and shortcomings

- Does it really scale up?
- Billions of pairs (depending on tokenization level)
- Experimental data set is too small to prove massive scale-up

Issues and shortcomings

- Does it really scale up?
- Billions of pairs (depending on tokenization level)
- Experimental data set is too small to prove massive scale-up

Possible Research Problems

- How to decrese the candidate blow-up?

Possible Research Problems

- How to decrese the candidate blow-up?
- storing the dictionary in some distributed key-value store?

Possible Research Problems

- How to decrese the candidate blow-up?
- storing the dictionary in some distributed key-value store?
- Exploiting the low number of candidates generated after map-phase?

Conclusion

- The problem of scale
- MapReduce is a nice paradigm for distributed large-scale jobs
- But we need specialized strategies

Questions?