Fast and Scalable Distributed Set Similarity Joins for Big Data Analytics

Chuitian Rong, Chunbin Lin, Yasin N. Silva, Jianguo Wang, Wei Lu, Xiaoyong Du

Arizona State University

Introduction

Set similarity join is an essential operation in big data analytics, e.g., data integration and data cleaning, that finds similar pairs from two collections of sets. Multiple techniques have been proposed to perform similarity joins using MapReduce in recent years.

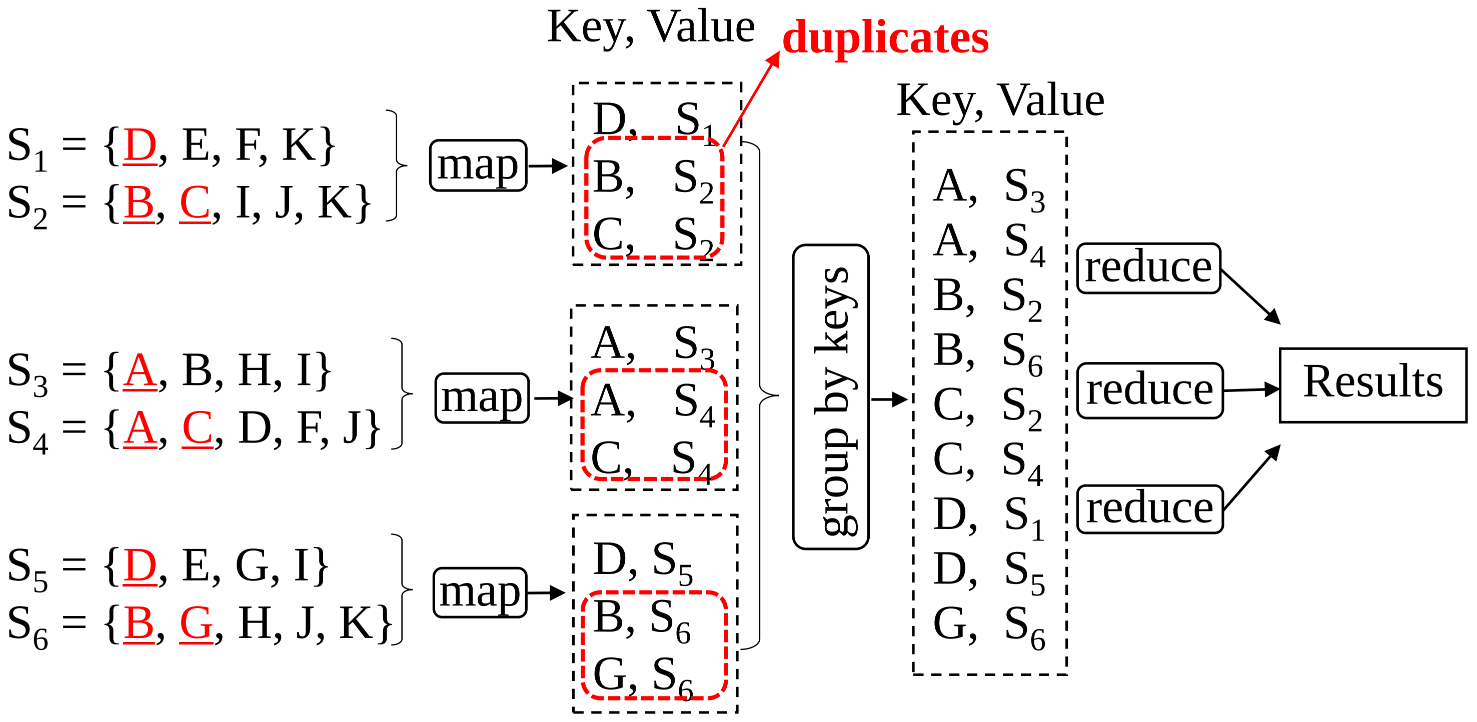
Existing techniques have several limitations.

- Generation of many duplicates
- Skewness problem
- Expensive verification processing

To address these problems, we have made the following contributions in our work:

- We proposed a vertical-partitioning based algorithm, called FS-Join, to support parallel set similarity joins without generating duplicates. In addition, it guarantees load balancing in both map and reduce phases.
- We introduced three new segment-based filtering methods, which significantly reduce the number of candidates.
- We proposed an optimization method by integrating horizontal data partitioning with vertical data partitioning to achieve higher scalability.

Existing Work



S_1 = {D, E, F, K}
S_2 = {B, C, I, J, K}
map
S_3 = {A, B, H, I}
S_4 = {A, C, D, F, J}
map
S_5 = {D, E, G, I}
S_6 = {B, G, H, J, K}
map

Key, Value
D, S_1
B, S_2
C, S_2
A, S_3
A, S_4
C, S_4
D, S_5
B, S_6
G, S_6

duplicates

group by keys

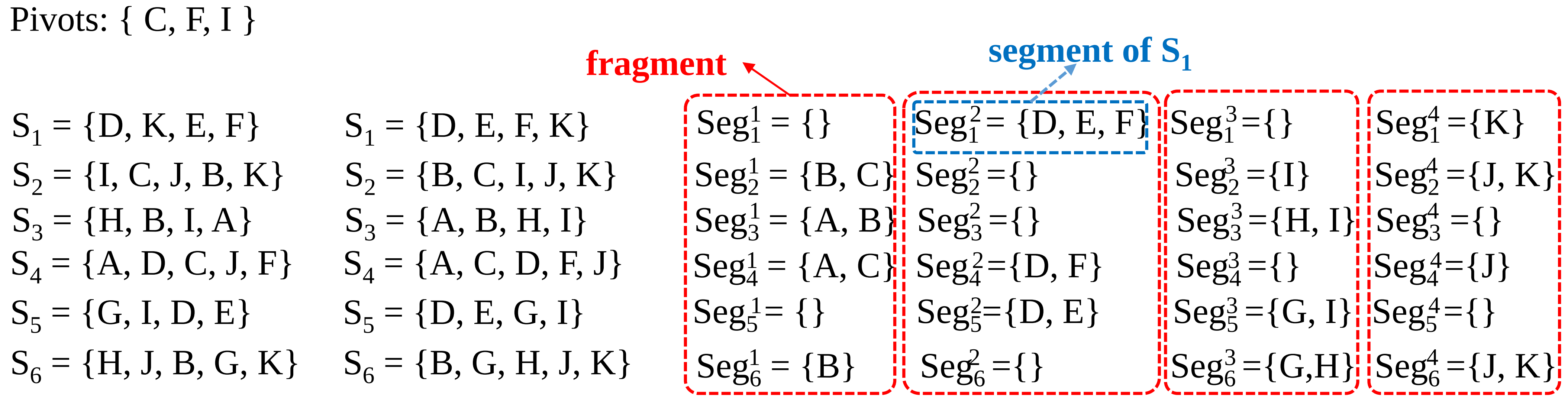
Key, Value
A, S_3
A, S_4
B, S_2
B, S_6
C, S_2
C, S_4
D, S_1
D, S_5
G, S_6

reduce
reduce
reduce
Results

Vertical Partitioning

Global Ordering: A → B → C → E → F → G → H → I → J → K

Pivots: { C, F, I }



fragment

segment of S_1

S_1 = {D, K, E, F}
S_2 = {I, C, J, B, K}
S_3 = {H, B, I, A}
S_4 = {A, D, C, J, F}
S_5 = {G, I, D, E}
S_6 = {H, J, B, G, K}

(a) Original sets

S_1 = {D, E, F, K}
S_2 = {B, C, I, J, K}
S_3 = {A, B, H, I}
S_4 = {A, C, D, F, J}
S_5 = {D, E, G, I}
S_6 = {B, G, H, J, K}

(b) Re-ordered sets

Seg_1^1 = {}	Seg_1^2 = {D, E, F}	Seg_1^3 ={}	Seg_1^4 ={K}
Seg_2^1 = {B, C}	Seg_2^2 ={}	Seg_2^3 ={I}	Seg_2^4 ={J, K}
Seg_3^1 = {A, B}	Seg_3^2 ={}	Seg_3^3 ={H, I}	Seg_3^4 ={}
Seg_4^1 = {A, C}	Seg_4^2 ={D, F}	Seg_4^3 ={}	Seg_4^4 ={J}
Seg_5^1 = {}	Seg_5^2 ={D, E}	Seg_5^3 ={G, I}	Seg_5^4 ={}
Seg_6^1 = {B}	Seg_6^2 ={}	Seg_6^3 ={G,H}	Seg_6^4 ={J, K}

(c) Partitioning based on pivots {C, F, I}

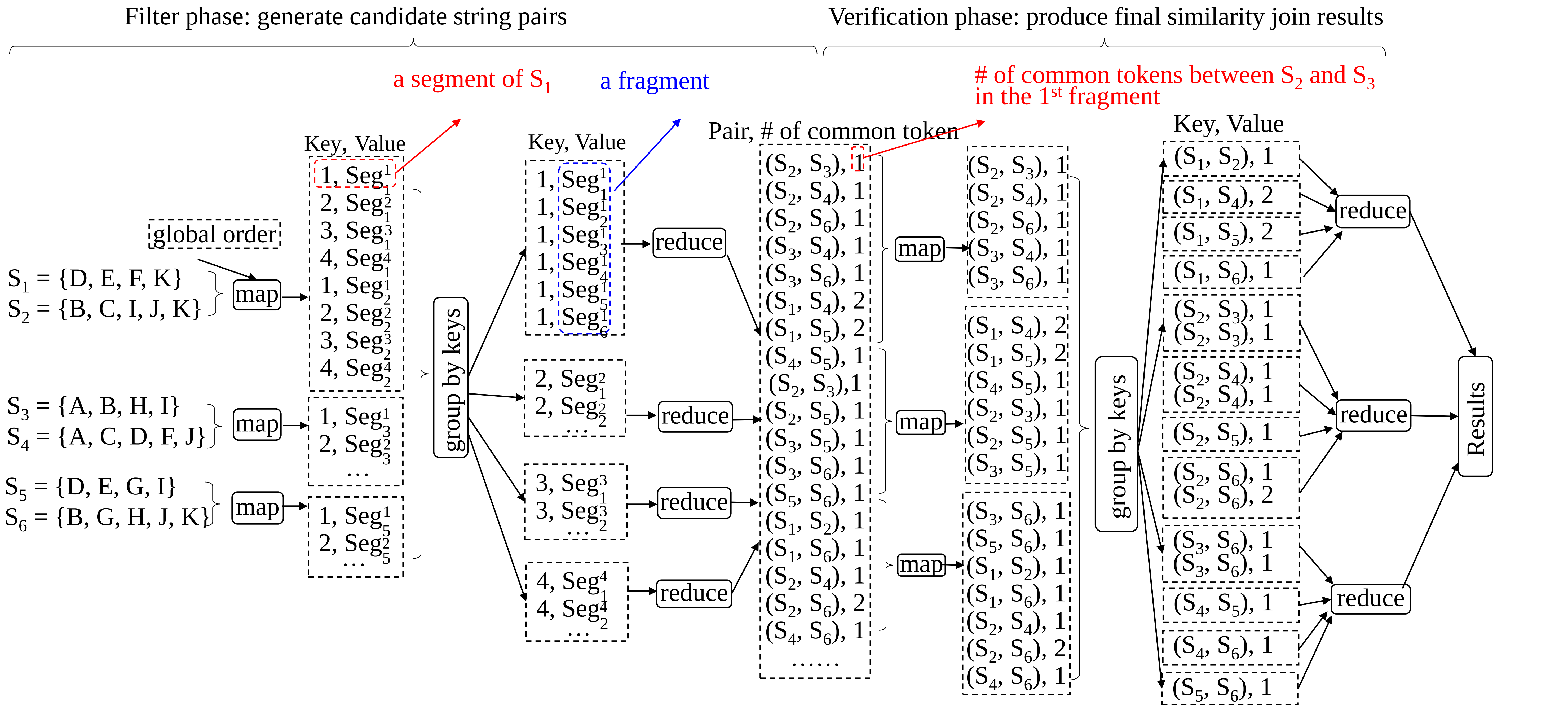
Pivot Selection

Random Selection (Random)
Even Interval (Even-Interval)
Even Token Frequency (Even-TF)

Filtering Methods

String Length Filtering(StrL-Filter)
Segment Length Filtering(SegL-Filter)
Segment Intersection Filtering(SegI-Filter)
Segment Difference Filtering(SegD-Filter)

Computation Framework of FS-Join



Filter phase: generate candidate string pairs

Verification phase: produce final similarity join results

a segment of S_1

a fragment

# of common tokens between S_2 and S_3 in the 1st fragment

global order

S_1 = {D, E, F, K}
S_2 = {B, C, I, J, K}
map
S_3 = {A, B, H, I}
S_4 = {A, C, D, F, J}
map
S_5 = {D, E, G, I}
S_6 = {B, G, H, J, K}
map

Key, Value
1, Seg_1^1
2, Seg_1^2
3, Seg_1^3
4, Seg_1^4
1, Seg_2^1
2, Seg_2^2
3, Seg_2^3
4, Seg_2^4

1, Seg_3^1
2, Seg_3^2
…

1, Seg_5^1
2, Seg_5^2
…

group by keys

Key, Value
1, Seg_1^1
1, Seg_2^1
1, Seg_3^1
1, Seg_4^1
1, Seg_5^1
1, Seg_6^1
reduce

2, Seg_1^2
2, Seg_2^2
…
reduce

3, Seg_1^3
3, Seg_2^3
…
reduce

4, Seg_1^4
4, Seg_2^4
…
reduce

Pair, # of common token
(S_2, S_3), 1
(S_2, S_4), 1
(S_2, S_6), 1
(S_3, S_4), 1
(S_3, S_6), 1
(S_1, S_4), 2
(S_1, S_5), 2
(S_4, S_5), 1
(S_2, S_3),1
(S_2, S_5), 1
(S_3, S_5), 1
(S_3, S_6), 1
(S_5, S_6), 1
(S_1, S_2), 1
(S_1, S_6), 1
(S_2, S_4), 1
(S_2, S_6), 2
(S_4, S_6), 1
……

map
(S_2, S_3), 1
(S_2, S_4), 1
(S_2, S_6), 1
(S_3, S_4), 1
(S_3, S_6), 1

map
(S_1, S_4), 2
(S_1, S_5), 2
(S_4, S_5), 1
(S_2, S_3), 1
(S_2, S_5), 1
(S_3, S_5), 1

map
(S_3, S_6), 1
(S_5, S_6), 1
(S_1, S_2), 1
(S_1, S_6), 1
(S_2, S_4), 1
(S_2, S_6), 2
(S_4, S_6), 1

group by keys

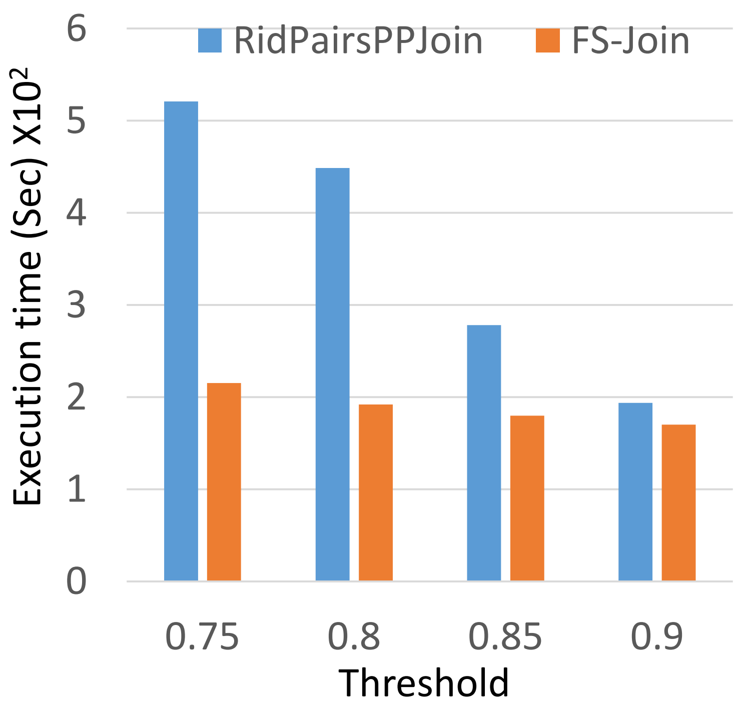
Key, Value
(S_1, S_2), 1
(S_1, S_4), 2
(S_1, S_5), 2
(S_1, S_6), 1
(S_2, S_3), 1
(S_2, S_3), 1
(S_2, S_4), 1
(S_2, S_4), 1
(S_2, S_5), 1
(S_2, S_6), 1
(S_2, S_6), 2
(S_3, S_6), 1
(S_3, S_6), 1
(S_4, S_5), 1
(S_4, S_6), 1
(S_5, S_6), 1

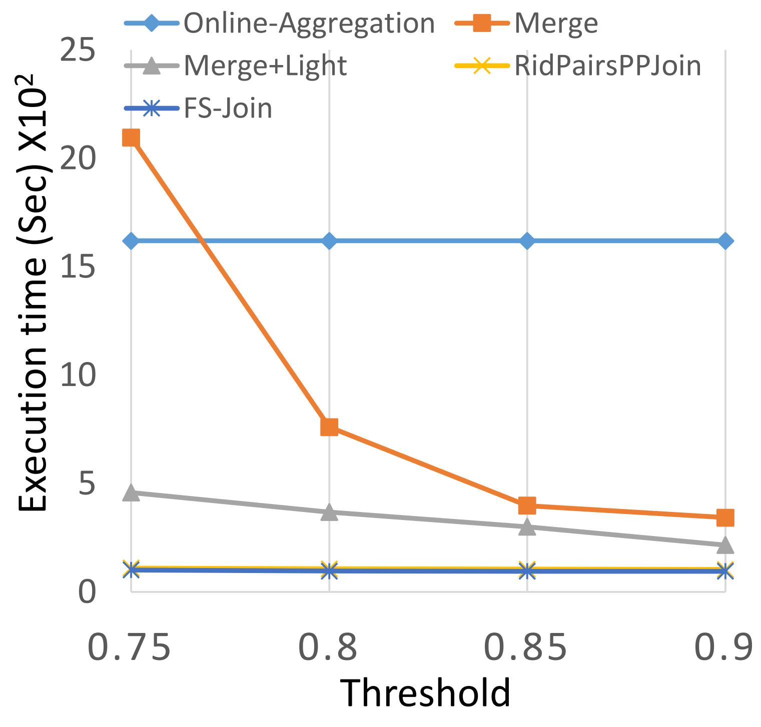
reduce
reduce
reduce
Results

Filters And Their Effectiveness

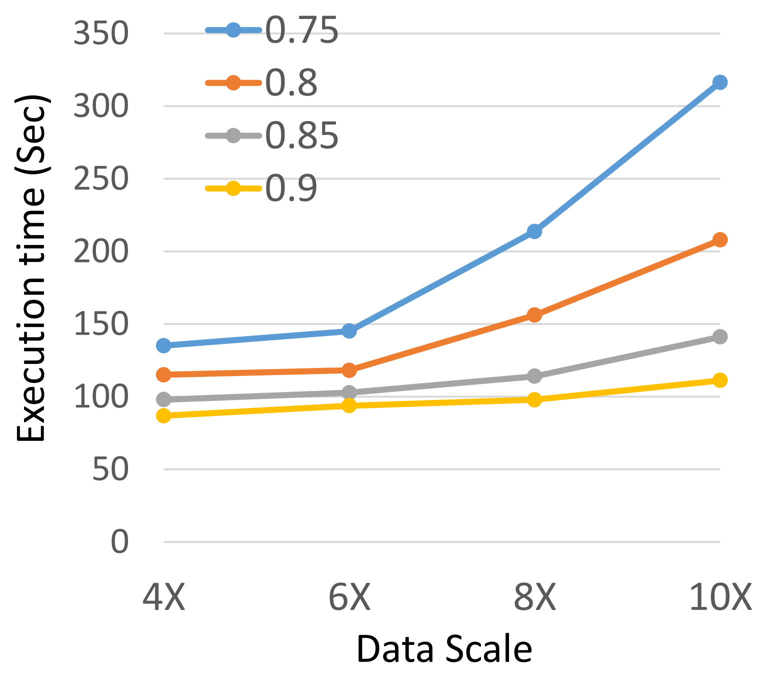
	Candidates Number		
Filter	Email(10%)	Wiki(1%)	PubMed(1%)
StrL	271,385,025	1,473,167,384	1,403,760,351
StrL + SegL	233,063,886	1,449,842,593	1,399,927,097
StrL + SegI	1,164,102	2,287,718	31,498
StrL + SegD	1,143,783	1,236,775	8,342
StrL + Prefix	1,011,428	1,147,016	792,185
All	493,644	515,664	6,840

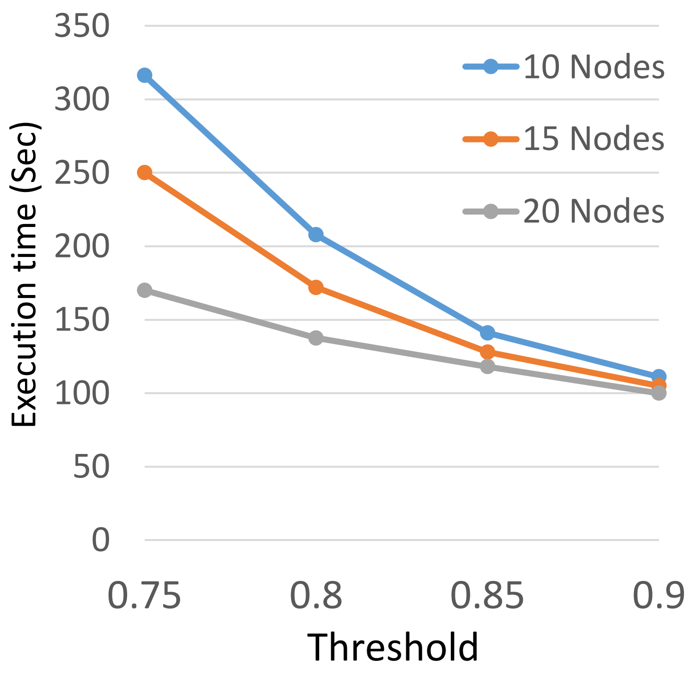
Comparison with Existing Methods





Scalability Tests


Execution time (Sec)
4X
6X
8X
10X
Data Scale


10 Nodes
15 Nodes
20 Nodes
Execution time (Sec)
Threshold