MapReduce-based Similarity Join for Metric Spaces

Yasin N. Silva, Jason M. Reed, Lisa M. Tsosie
Arizona State University

Slides prepared for Cloud-I workshop in VLDB 2012.

Overview

- Motivation
- Algorithm
- Implementation
- Performance Evaluation
- Conclusions and Future Work

Introduction

- Similarity Joins used by many companies
- Internet companies have massive amounts of data
- Many non-distributed approaches to Similarity Join problem
- Few cloud based approaches

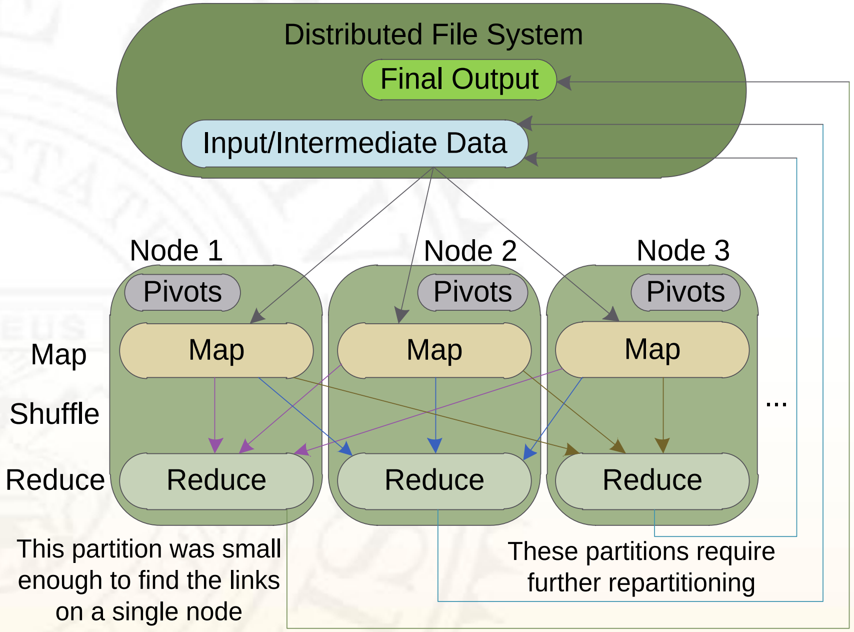
Our Contribution

- MRSimJoin Algorithm
- General enough for any data in metric space
- Guidelines to implement in Hadoop
- Evaluation of performance and scalability
- Evaluation of pivot numbers, means of choosing a good number of pivots

Overview

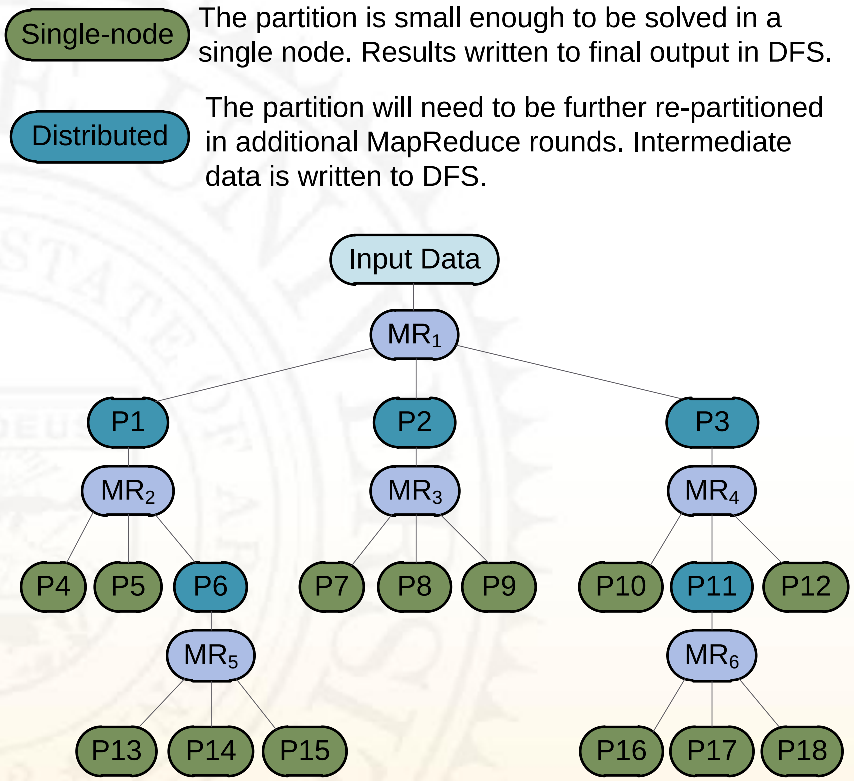
- Motivation
- Algorithm
- Implementation
- Performance Evaluation
- Conclusions and Future Work

MRSimJoin Round



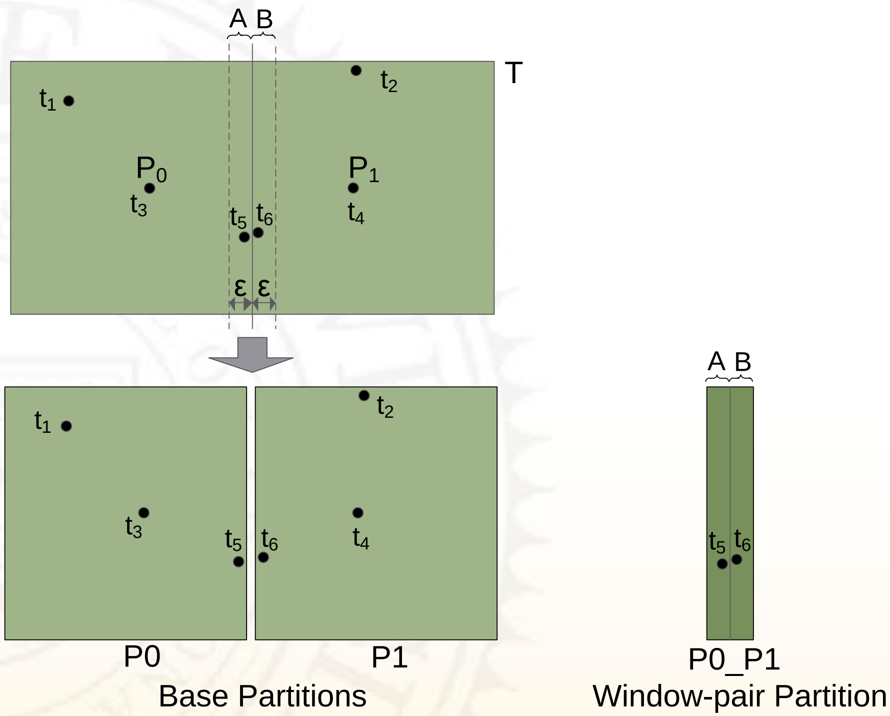
- MRSimJoin iteratively partitions the data
  - If partition is small enough, solve in single node SJ routine
- The process is divided into a sequence of rounds
- The initial round partitions the input data
- Any subsequent round repartitions a previously generated partition

Multiple Rounds

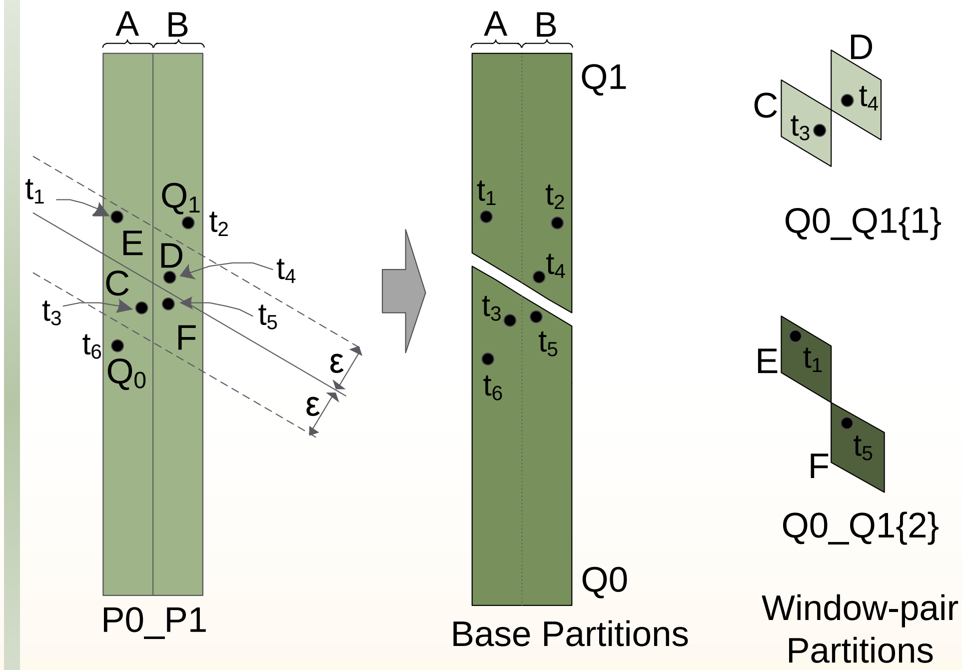


- Each round corresponds to a MapReduce job
- The output of a round includes:
  1. Result links for the small partitions that were processed in a single-node
  2. Intermediate data for partitions that require further partitioning

Partitioning Data

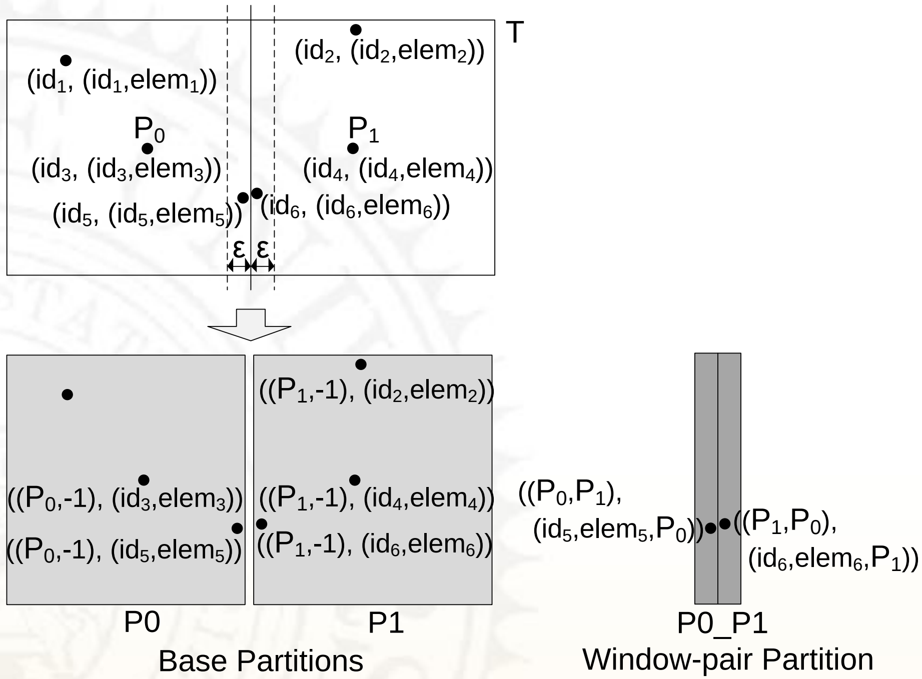


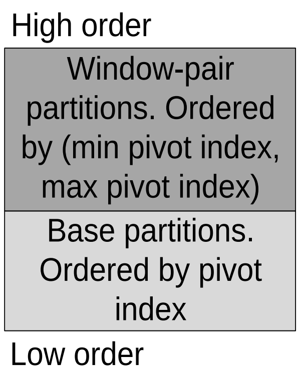
Partitioning a Base Partition



Partitioning a Window-Pair Partition

Partition a Base Set





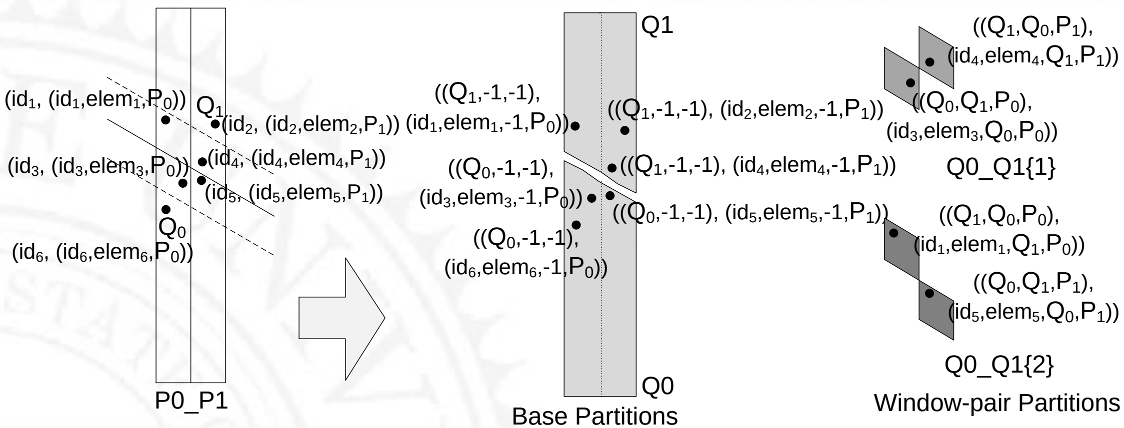
(a) General order of partitions

P0_P1	((P_0,P_1), (id_5,$elem_5$,P_0)) ((P_1,P_0), (id_6,$elem_6$,P_1))
P1	((P_1,-1), (id_2,$elem_2$)) ((P_1,-1), (id_4,$elem_4$)) ((P_1,-1), (id_6,$elem_6$))
P0	((P_0,-1), (id_1,$elem_1$)) ((P_0,-1), (id_3,$elem_3$)) ((P_0,-1), (id_5,$elem_5$))

(b) Order of partitions with 2 pivots

- Choose pivots (randomly chosen subset of input data)
- Create base partitions around closest pivots
- Create window-pair partitions between partitions
- Each partition is sent to a reduce group

Partition a Window-Pair Set



- Choose pivots
- Partition data around pivots
- Create windows space between base partitions
  - The window of a window is aware of previous partitioning

High order
Window-pair partitions. Ordered by (min pivot index, max pivot index, sequence)
Base partitions. Ordered by pivot index
Low order

(a) General order of partitions

Q0_Q1{2}	((Q$_1$,Q$_0$,P$_0$), (id$_1$,elem$_1$,Q$_1$,P$_0$)) ((Q$_0$,Q$_1$,P$_1$), (id$_5$,elem$_5$,Q$_0$,P$_1$))
Q0_Q1{1}	((Q$_1$,Q$_0$,P$_1$), (id$_4$,elem$_4$,Q$_1$,P$_1$)) ((Q$_0$,Q$_1$,P$_0$), (id$_3$,elem$_3$,Q$_0$,P$_0$))
Q1	((Q$_1$,-1,-1), (id$_1$,elem$_1$,-1,P$_0$)) ((Q$_1$,-1,-1), (id$_2$,elem$_2$,-1,P$_1$)) ((Q$_1$,-1,-1), (id$_4$,elem$_4$,-1,P$_1$))
Q0	((Q$_0$,-1,-1), (id$_3$,elem$_3$,-1,P$_0$)) ((Q$_0$,-1,-1), (id$_6$,elem$_6$,-1,P$_0$)) ((Q$_0$,-1,-1), (id$_5$,elem$_5$,-1,P$_1$))

(b) Order of partitions with 2 pivots

Overview

- Motivation
- Algorithm
- Implementation
- Performance Evaluation
- Conclusions and Future Work

Implementation

- Generic enough to implement in any MR framework
- Hadoop implementation:
  - Distribution of Atomic parameters
    - Uses jobConf
  - Distribution of pivots
    - Uses Distributed Cache
  - Renaming Directories
    - Renaming directories does not move data

Overview

- Motivation
- Algorithm
- Implementation
- Performance Evaluation
- Conclusions and Future Work

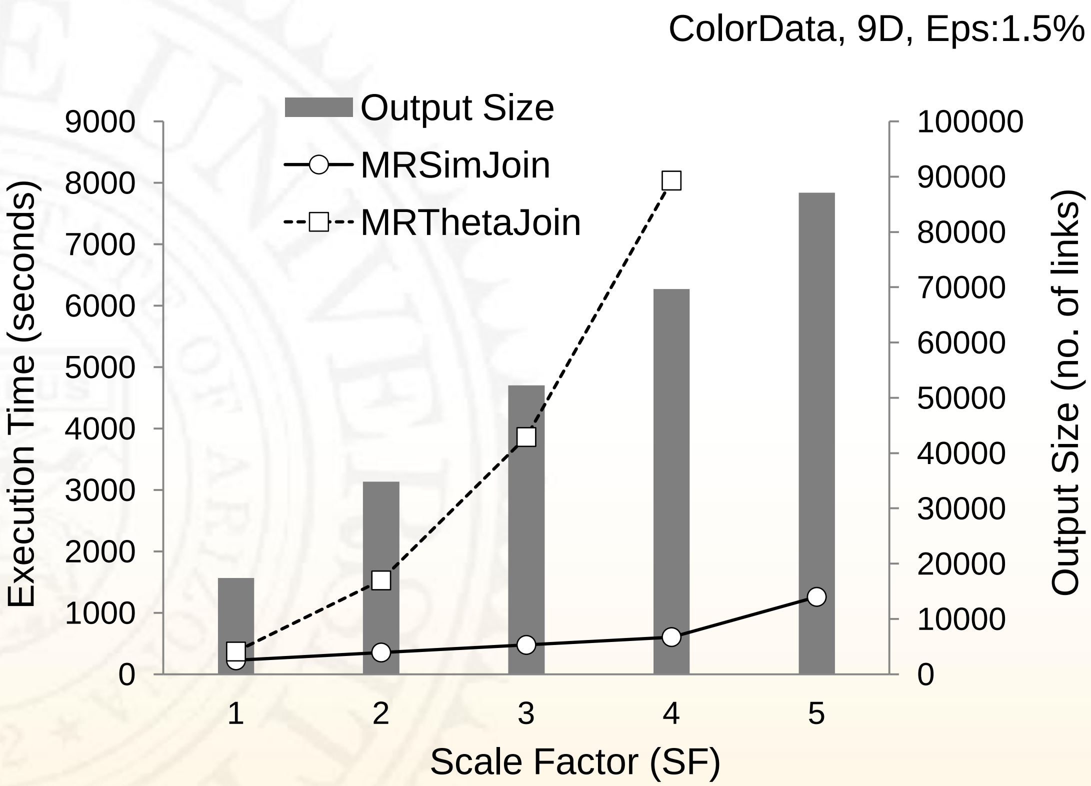
Testing Platform

- Amazon EC2 Cloud
  - 4 virtual cores (2 EC2 Compute Units each)
  - 15 GB memory
  - 1,690 GB local storage
  - 64 bit platform
- Hadoop 0.20.2
  - 64 MB block size
  - 10 nodes (1 master, 9 worker nodes)
- Memory Threshold
  - 32 MB

Test Data Information

- SynthData
  - Synthetic Data set
  - 16D Data
  - Vector components values are [0-1000]
  - Scale Factor 1 = 5 million records
- ColorData
  - Corel Color Moments Dataset
  - 9D Data
  - Vector components values are [-4.8 – 4.4]
  - Scale Factor 1 = 5 million records

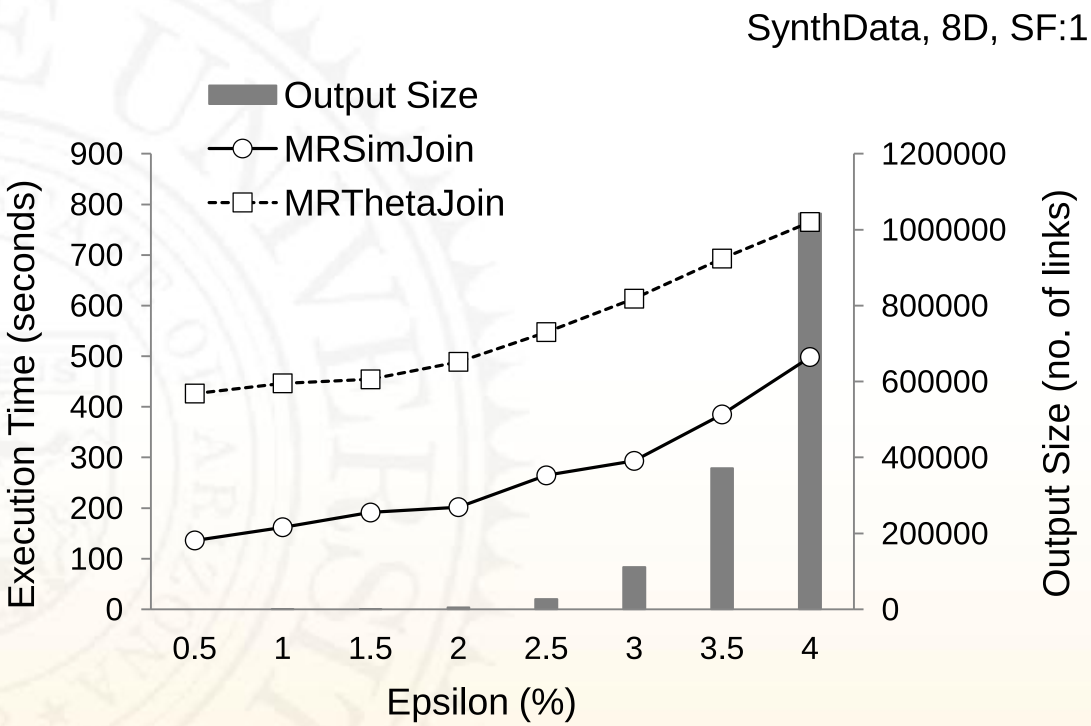
Increasing Scale Factor



- ColorData
  - SF1-SF5
  - 9D Color Vectors
  - Epsilon 1.5%
  - 1.6x (SF1) – 13.3x (SF4)
- SynthData
  - SF1-SF5
  - 8D Vectors
  - Epsilon 1.5%
  - 2.4x (SF1) – 11.4x (SF3)

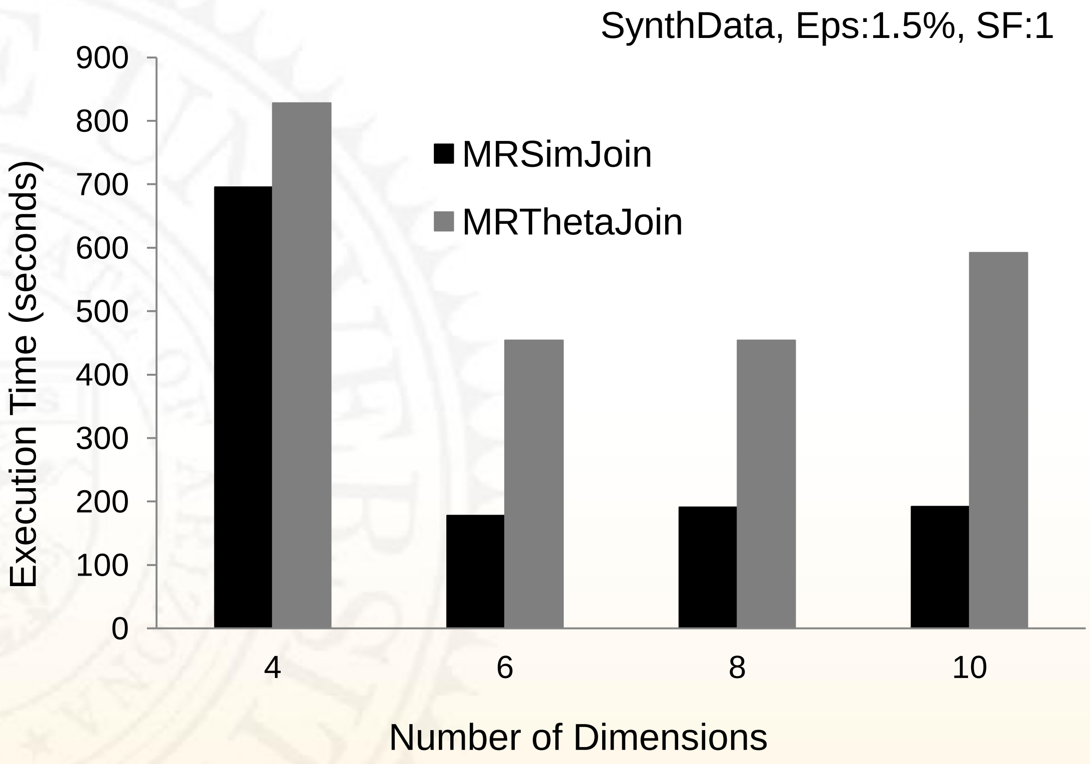
MR ThetaJoin: A. Okcan et. al.. Processing Theta-Joins using MapReduce, SIGMOD 2011

Increasing Epsilon



- SynthData
  - 0.5% - 4.0% epsilon
  - 8D Vectors
  - SF1
  - 1.4x – 3x
- ColorData
  - 0.5% - 2.5% Epsilon
  - 9D Color Vectors
  - SF1
  - ~60% of MRThetaJoin

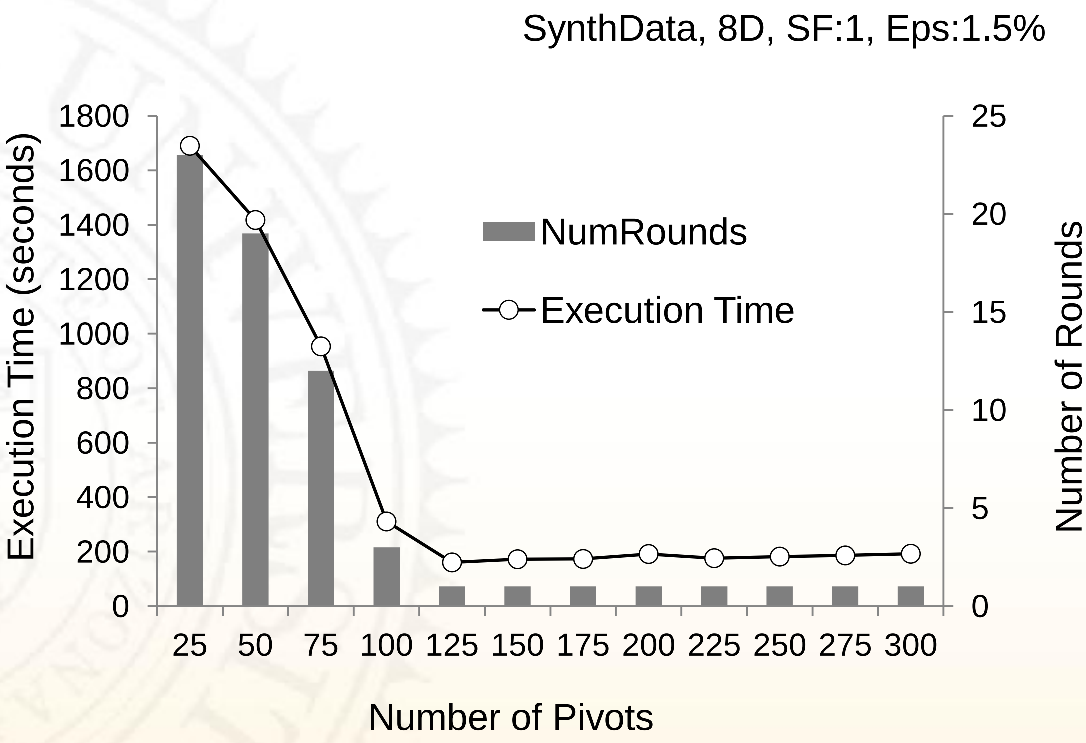
Increasing Dimension



- SynthData
  - 4D-10D
  - Epsilon 1.5%
  - SF1
  - MRThetaJoin is 20 - 200% higher

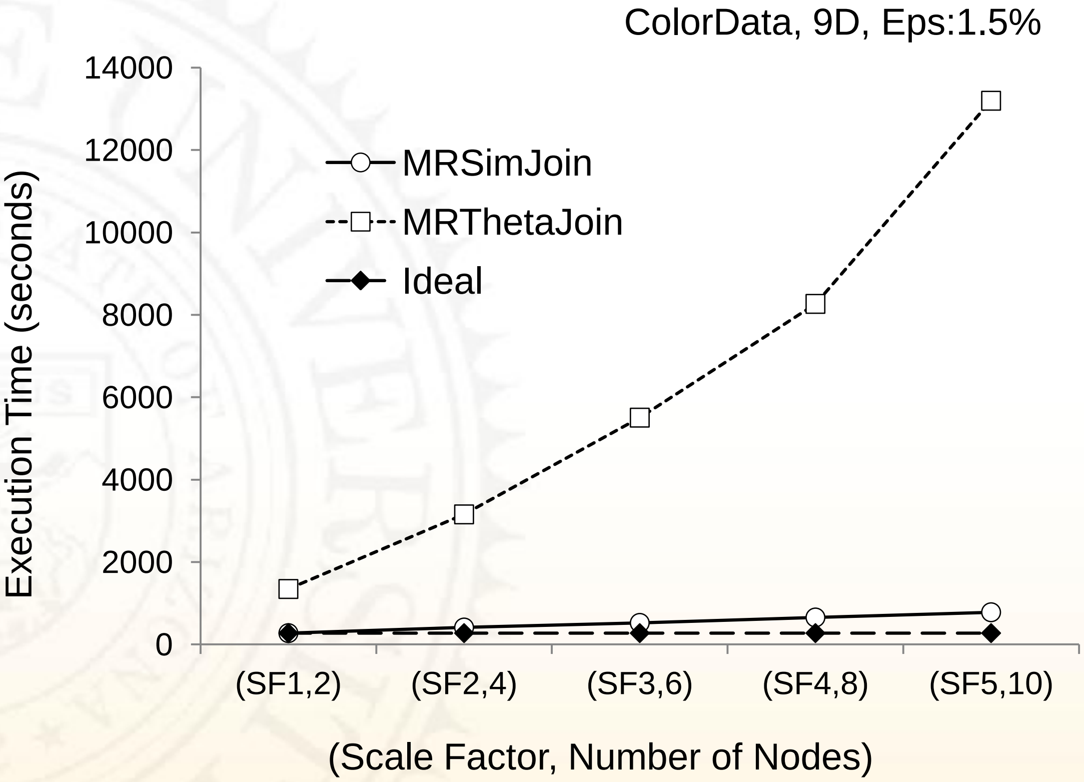
ARIZONA STATE UNIVERSITY

Increasing Pivot Number



- SynthData
  - 25 -300 pivots
  - 8D vector
  - Epsilon 1.5%

Increasing Node Number & SF



- ColorData
  - (SF1, nodes) – (SF5,10 nodes)
  - 9D Color Vector
  - Epsilon 1.5%
  - MRThetaJoin
    - 9.8x between (SF1, 2) & (SF5, 10)
  - MRSimJoin
    - 2.8x between (SF1, 2) & (SF5, 10)

Overview

- Motivation
- Algorithm
- Implementation
- Performance Evaluation
- Conclusions and Future Work

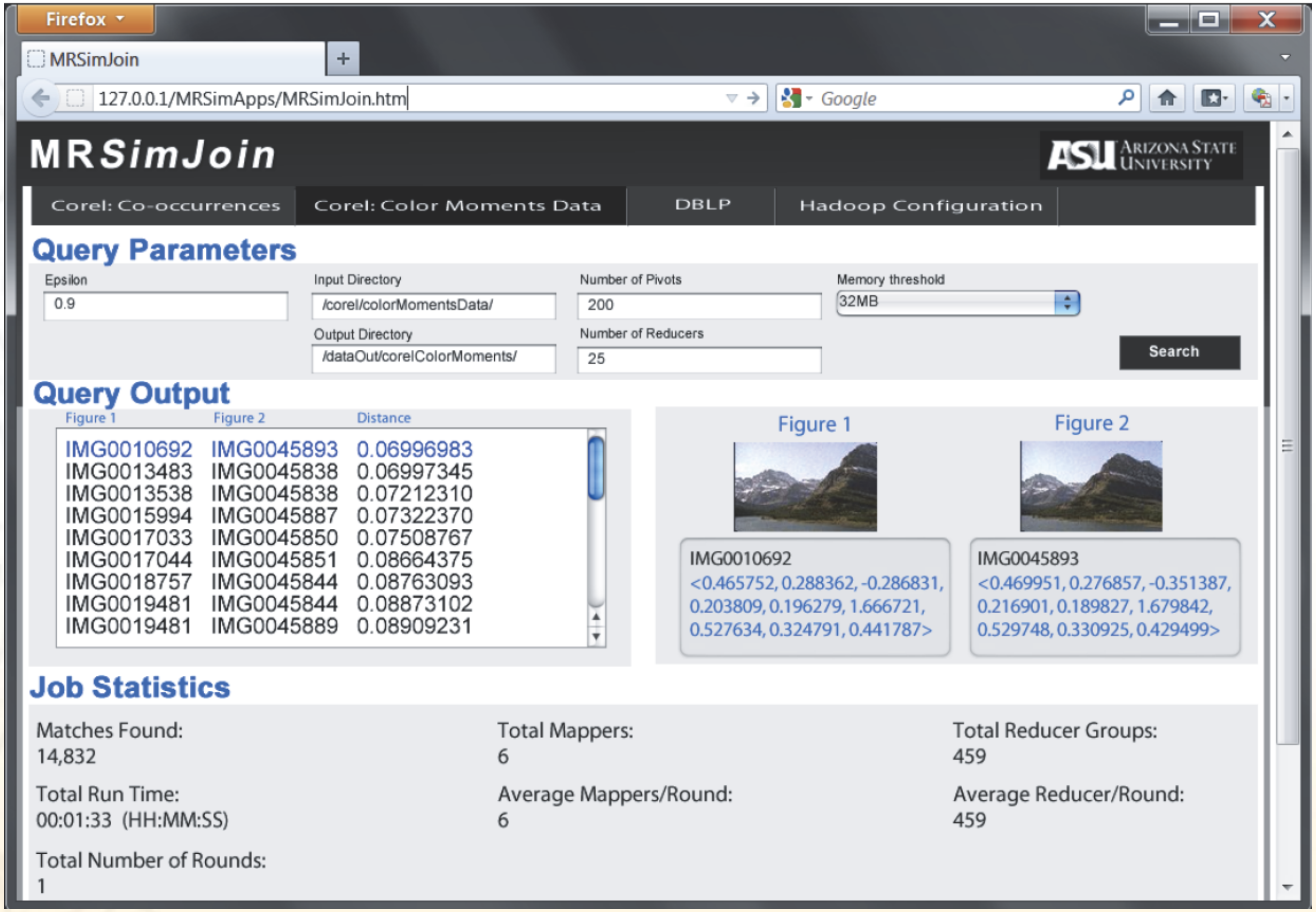
Conclusions

- MRSimJoin efficiently solves the distributed Similarity Join problem
- Significantly better than state-of-the-art MapReduce arbitrary join algorithm
- Partitions data till data can be joined in single node
- Any data set that lies in a metric space
- Scalable
- Highly parallel

Future Work

- Other similarity-aware operators
  – kNN Join, kDistance Join, etc
- Indexing techniques for implementing Similarity Join operations
- Cloud queries with multiple similarity-based operators

Questions?



Firefox
MRSimJoin
127.0.0.1/MRSimApps/MRSimJoin.htm
Google
MRSimJoin
ASU Arizona State University
Corel: Co-occurrences
Corel: Color Moments Data
DBLP
Hadoop Configuration
Query Parameters
Epsilon
0.9
Input Directory
/corel/colorMomentsData/
Output Directory
/dataOut/corelColorMoments/
Number of Pivots
200
Number of Reducers
25
Memory threshold
32MB
Search
Query Output
Figure 1
Figure 2
Distance
IMG0010692 IMG0045893 0.06996983
IMG0013483 IMG0045838 0.06997345
IMG0013538 IMG0045838 0.07212310
IMG0015994 IMG0045887 0.07322370
IMG0017033 IMG0045850 0.07508767
IMG0017044 IMG0045851 0.08664375
IMG0018757 IMG0045844 0.08763093
IMG0019481 IMG0045844 0.08873102
IMG0019481 IMG0045889 0.08909231
Figure 1
Figure 2
IMG0010692
<0.465752, 0.288362, -0.286831, 0.203809, 0.196279, 1.666721, 0.527634, 0.324791, 0.441787>
IMG0045893
<0.469951, 0.276857, -0.351387, 0.216901, 0.189827, 1.679842, 0.529748, 0.330925, 0.429499>
Job Statistics
Matches Found:
14,832
Total Run Time:
00:01:33 (HH:MM:SS)
Total Number of Rounds:
1
Total Mappers:
6
Average Mappers/Round:
6
Total Reducer Groups:
459
Average Reducer/Round:
459