An Experimental Survey of MapReduce-based Similarity Joins

Faculty: Yasin Silva Participants: Jason Reed, Kyle Brown, A.J. Wadsworth, Chuitian Rong, Nathan Middlebrook

Arizona State University

Motivation

The Problem

- Big-Data systems have been introduced to efficiently process and analyze massive amounts of data.
- One of the key data processing and analysis operations is the Similarity Join (SJ), which finds similar pairs of objects of two datasets.
- Several SJ techniques for Big-Data have been proposed but most of these techniques were developed in parallel and thus did not compare with alternative approaches.
- Consequently, there is not a clear understanding of how these techniques compare to each other and which technique to use in specific scenarios.

Our Contribution

- The core goal of the proposed project is to address this problem by focusing on the study, classification and benchmarking of all the SJ techniques proposed for Big-Data systems.
- Open source implementation of all the algorithms using the Hadoop Map-Reduce platform (consider the main framework to process Big Data).

Types of Experimental Tests

- Increasing distance threshold (ε)
- Increasing data size (scale factor)
- Increasing number of cluster nodes and data size

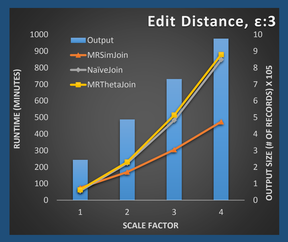
For String data:

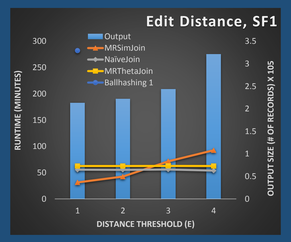
- Increasing length of strings
- Increasing alphabet size
- Variable string length Vs fixed length

For Vector data:

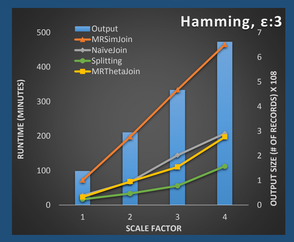
- Increasing dimensionality

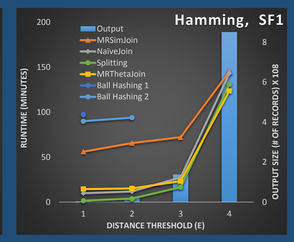
Edit Distance



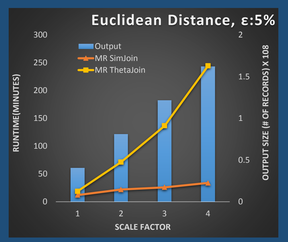


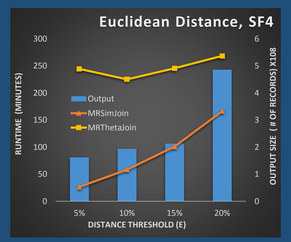
Hamming Distance



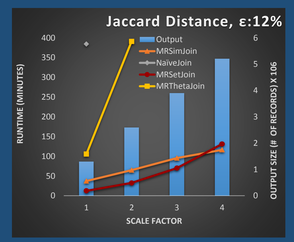


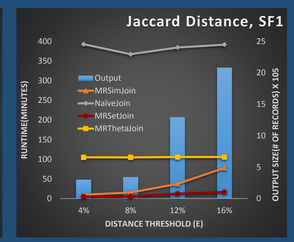
Euclidean Distance





Jaccard Distance





Classification of Algorithms

Algorithm	Supported Distance/ Similarity Functions	Supported Data Types			
		Text/String	Numeric	Vector	Set
Naïve Join	Any DF	●	●	*	●
Ball Hashing 1	Hamming Distance Edit Distance	●			
Ball Hashing 2	Hamming Distance Edit Distance	●			
Subsequence	Edit Distance	●			
Splitting	Hamming Distance Edit Distance	●			
Hamming Code	Hamming Distance	●			
Anchor Points	Hamming Distance Edit Distance	●	*	*	
MRThetaJoin	Any DF	●	●	●	●
MRSimJoin	Any metric DF	●	●	●	●
MRSetJoin	JS, TC, CC, Edit Distance*	*			●
Online Aggregation	JS, RS, DS, SC, VC				●
Lookup	JS, RS, DS, SC, VC				●
Sharding	JS, RS, DS, SC, VC				●

● Natively Supported

* Can be extended to support this data type or distance function

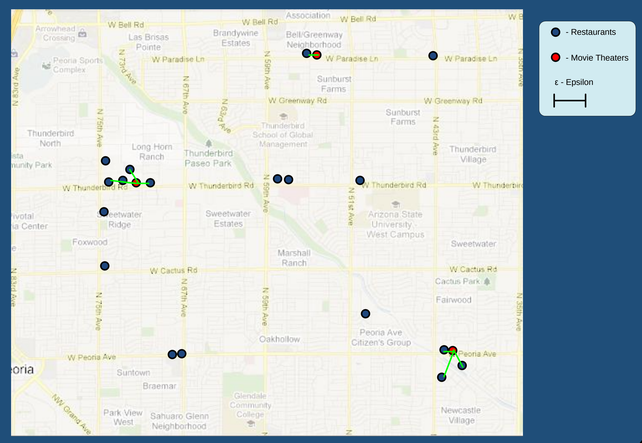
JS=Jaccard Similarity, TC=Tanimoto Coefficient, CC=Cosine Coefficient, RS=Ruzicka Similarity, DS=Dice Similarity, SC=Set Cosine Sim., VC=Vector Cosine Sim.

Dataset

- The experiments used a slightly modified version of the Harvard bibliographic dataset.
- Each record contains the following attributes: unique ID, title, date issued, record change date, record creation date, Harvard record ID, first author name, all author names, and vector.
- The dataset for scale factor 1 (SF1) contains 200K records.
- The datasets for SF greater than 1 were generated in such a way that the number of matches (links) of any SJ operation in SF*N* is *N* times the number of links in SF1.
- For vector data, the datasets for higher SF were obtained adding shifted copies of the SF1 dataset where the distance between copies were greater than the maximum value of ε.

Similarity Join Example

Goal: Find pairs of restaurants and movie theaters that are close to each other. "close" = within ε of one another



This research project is supported by an Amazon Research grant.