Entity Matching for Semistructured Data in the Cloud

Marcus Paradies
IBM F2CE Workshop
December 1, 2011
Outline

1. Motivation

2. Entity Matching

3. MAXIM: Entity Matching in the Cloud

4. Summary
Motivation

Enriching/Improving Wikipedia

References from Wikipedia article Hash join

References


External links

Motivation

Enriching/Improving Wikipedia

Lookup in the CiteSeer database

CiteSeer

No results found

Your search ""An Adaptive Hash Join Algorithm for Multiuser Environments"" did not match any documents.
Please try a different search.
Motivation

Enriching/Improving Wikipedia

Lookup in Google

An Adaptive Hash Join Algorithm for Multiuser Environments

Ungefähr 830.000 Ergebnisse (0,44 Sekunden)

Meinten Sie: An Adaptive Hash Join Algorithm for Multi User Environments

Wissenschaftliche Artikel zu An Adaptive Hash Join Algorithm for Multiuser Environments

... hash join algorithm for multiuser environments - Zeller - Zitiert durch: 85
Adaptive query processing: Technology in evolution - Hellerstein - Zitiert durch: 230
The cougar approach to in-network query processing in ... - Yao - Zitiert durch: 922

[PDF] An Adaptive Hash Join Algorithm for Multiuser Environments
citesee.x.ist.psu.edu/.../download?doi=10... - Diese Seite übersetzen  +7
Dateiformat: PDF/Adobe Acrobat - Schnellansicht
von H Zeller - Zitiert durch: 85 - Ähnliche Artikel
An Adaptive Hash Join Algorithm for Multiuser Environments. Hansjörg Zeller, Jim Gray. TANDEM. 10100 N. Tantau Avenue, LOC 251-05. Cupertino, CA 95014 ...
Motivation

Wikipedia in a nutshell

Characteristics

- 3.7 Mio articles (english Wikipedia database)
- Dataset size about 30GB of XML (without history)
- 3.6 Mio references
- References are categorized into books, journals, websites, etc.
Motivation

Wikipedia in a nutshell

Characteristics

- 3.7 Mio articles (english Wikipedia database)
- Dataset size about 30GB of XML (without history)
- 3.6 Mio references
- References are categorized into books, journals, websites, etc.

Challenges

- Articles in Wikipedia are incomplete
- Articles in Wikipedia are inaccurate
- Articles in Wikipedia are subjective
Motivation

Problem Statement

Definition

Given two datasets of records, $R$ and $S$, a set of attributes $a_1, \ldots, a_n$, a set of similarity functions $sim_{a_1}, \ldots, sim_{a_n}$ and a similarity threshold $\tau$, the task between $R$ and $S$ is defined as finding and combining all pairs of records from $R$ and $S$ where

$$\sum_{i=1}^{n} sim_{a_i}(R.a_i, S.a_i) \geq \tau$$

---

Wikipedia Data set

CiteSeer Data set
Given two datasets of records, $R$ and $S$, a set of attributes $a_1, \ldots, a_n$, a set of similarity functions $sim_{a_1}, \ldots, sim_{a_n}$ and a similarity threshold $\tau$, the task between $R$ and $S$ is defined as finding and combining all pairs of records from $R$ and $S$ where $\sum_{i=1}^{n} sim_{a_i}(R.a_i, S.a_i) \geq \tau$.  

{{Cite book
  | last = Mumford
  | first = David
  | authorlink = David Mumford
  | title = The Red Book of Varieties and Schemes
  | publisher = [[Springer]]
  | location = Berlin
  | date = 1999
  | page = 198
  | isbn = 354063293X
}}

Wikipedia Data set

CiteSeer Data set
Entity Matching
Entity Matching

What is Entity Matching?

- Entity Matching has quadratic runtime behavior
- Entity Matching has high CPU- and memory demands
- The definition of "what is similar" is domain-dependent
What is Entity Matching?

Challenges

- Entity Matching has quadratic runtime behavior
- Entity Matching has high CPU- and memory demands
- The definition of “what is similar” is domain-dependent
Entity Matching

Entity Matching Architecture

Data Source $S_1$

Data Source $S_2$

Blocking

$b_1$

$b_2$

$b_3$

$\cdots$

$b_n$

Matching

Match Result $R$

Entity Matching for Semistructured Data in the Cloud

Marcus Paradies
Entity Matching

Entity Matching Architecture

Data Source $S_1$

Data Source $S_2$

Blocking

$b_1$

$b_2$

$b_3$

$\vdots$

$b_n$

Matching

Match Result $R$

How can we improve the runtime of an EM task?
Entity Matching

Entity Matching Architecture

Data Source $S_1$

Data Source $S_2$

Blocking

Matching

Match Result $R$

Distributed Blocking

$b_1$

$b_2$

$b_3$

$\ldots$

$b_n$
Entity Matching

Entity Matching Architecture

Data Source $S_1$

Data Source $S_2$

Blocking

$b_1$

$b_2$

$b_3$

$\vdots$

$b_n$

Matching

Match Result $R$

Distributed Blocking

Parallel Matching

Marcus Paradies

Entity Matching for Semistructured Data in the Cloud
MAXIM: Entity Matching in the Cloud
MAXIM: Entity Matching in the Cloud

Requirements and Approach

Requirements

- Efficient processing of semistructured data
- Scalability to large datasets
- Independency from specific similarity functions
- Ability to easily add new similarity functions

Main Idea

- Use MapReduce and ChuQL to process semistructured data
- Use a search-based blocking to generate candidate pairs
- Apply similarity functions to candidate pairs within a block
Requirements and Approach

Requirements

- Efficient processing of semistructured data
- Scalability to large datasets
- Independency from specific similarity functions
- Ability to easily add new similarity functions

Main Idea

- Use MapReduce and ChuQL to process semistructured data
- Use a search-based blocking to generate candidate pairs
- Apply similarity functions to candidate pairs within a block
ChuQL by example

Wordcount in ChuQL

```chuql
1  mapreduce {
2    input { fn:collection("hdfs://wiki/") }
3    rr { for $rev in $hxml:in//revision
4        return{"key": fn:data($x//username|$x//ip),
5            "val": $x//title } }
6    map { $hxml:in }
7    reduce { {"key": $hxml:in=>"key", "value": fn:count($hxml:in=>"val")} }
8    rw { <author name="{$hxml:in=>"key"}" count="{$hxml:in=>"val"}"/> }
9    output { fn:put("hdfs://outputdir/") }
10  }
```
MAXIM: Entity Matching in the Cloud

Architecture

- Hadoop cluster with up to 40 nodes
- Each node runs a search engine and an attached full-text index
- Each node runs an in-memory XQuery processor
- Semistructured data is partitioned and placed on HDFS
MAXIM: Entity Matching in the Cloud

Processing Stages

Three Stages

- Preparation Stage
- Blocking Stage
- Matching Stage
**Stage 1: Preparation Stage**

- Extracts references from Wikipedia
- Reads and transforms records from CiteSeerX
- Sends CiteSeerX data to local full-text index
MAXIM: Entity Matching in the Cloud

Processing Stages

Stage 2: Blocking Stage

- Reads extracted references from HDFS
- Probes full-text index to retrieve candidate publications
- Assign candidate publications to block(s)
**Stage 3: Matching Stage**

- Read blocks from HDFS
- Generate candidate pairs and apply similarity functions
- Store matching pairs and their similarity
Stage 1: Preparation Stage

Extracting References

External links

Stage 1: Preparation Stage

Extraction References

External links [edit]


Indexing Publications

Entity Matching for Semistructured Data in the Cloud

Marcus Paradies
MAXIM: Entity Matching in the Cloud

Stage 1: Preparation Stage

Extracting References

Extraction

Transformation

Indexing Publications

Entity Matching for Semistructured Data in the Cloud

Marcus Paradies
Stage 1: Preparation Stage

Extracting References

Indexing Publications

HDFS
MAXIM: Entity Matching in the Cloud

Stage 1: Preparation Stage

Extracting References

Indexing Publications

HDFS
Read and Transformation

Entity Matching for Semistructured Data in the Cloud

Marcus Paradies
Stage 1: Preparation Stage

Extracting References

Indexing Publications

External links


<reference type="journal">
  | author1 = Hansjörg Zeller
  | author2 = Jim Gray
  | title = An Adaptive Hash Join Algorithm for Multi-User Environments
  | journal = Proceedings of the 16th VLDB conference
  | year = 1990
  | pages = 186–197
</reference>

Lucene
Index
HDFS
Read and Transformation
Indexing

Entity Matching for Semistructured Data in the Cloud

Marcus Paradies

14 / 18
Stage 2: Blocking Stage

Block generation

- Each reference generates a set of candidate publications
- Each candidate publication is inserted into all blocks, which are listed in reference
Stage 2: Blocking Stage

Block generation

- Each reference generates a set of candidate publications
- Each candidate publication is inserted into all blocks, which are listed in reference

Example
Stage 2: Blocking Stage

Distributed Search in MAXIM

(a) Send HTTP request (query)
(b) HTTP response (partial result)
(c) Collect partial results

Node 1
   Data Node
   Task Tracker
   ChuQL Engine

Node 2
   Search Engine
   Full-text Index

Node 3
   Search Engine
   Full-text Index

Node 4
   Search Engine
   Full-text Index

Node 5
   Search Engine
   Full-text Index
Stage 3: Matching Stage

- Applies user-defined similarity functions to candidate pairs
- Each attribute can be evaluated by a specific similarity function
Stage 3: Matching Stage

- Applies user-defined similarity functions to candidate pairs
- Each attribute can be evaluated by a specific similarity function

Number of candidate pairs

\[ CP = \sum_{i=1}^{n} C_i \times R_i \]  

- \( n \) - \# of blocks in \( B_1, \ldots, B_n \)
- \( R_i \) - \# of references in block \( B_i \)
- \( C_i \) - \# of candidate publications in block \( B_i \)
- \( CP \) - \# of candidate pairs to verify
Wikipedia provides many opportunities for research

Need for efficiently processing semistructured data is increasing

Entity Matching is critical for data integration and data cleaning

Entity Matching is difficult to parallelize due to unbalanced data partitions

MAXIM parallelizes EM by building blocks of similar records in a classification fashion

MAXIM allows to define own similarity functions and computation functions without changing the algorithm
“Everything that can be invented has been invented.”

(Charles H. Duell, Commissioner, U.S. Office of Patents, 1899)
Experiments

Scaleup and Speedup

(a) Speedup for all stages

(b) Scaleup for preparation stage
Experiments

Query Performance

Figure: Query Performance for different result set sizes and cluster sizes.

Marcus Paradies

Entity Matching for Semistructured Data in the Cloud
Experiments

Blocking Accuracy

Figure: Blocking accuracy for different typographical error classes.
Experiments

Number of Candidate Pairs

Figure: Number of candidate pair verifications in the matching stage.