SPONSORED BY THE
Federal Ministry
of Education
and Research

Freie Universität Berlin

Cluster Mechanisms in a Self-Organizing Distributed Semantic Store

Marko Harasic, Anne Augustin, Robert Tolksdorf, Philipp Obermeier
{harasic,aaugusti,tolk,obermeie}@inf.fu-berlin.de

Web Based Information Systems
http://digipolis.ag-nbi.de

Overview

- Motivation
- Algorithms
- Evaluation
- Conclusion & Future work

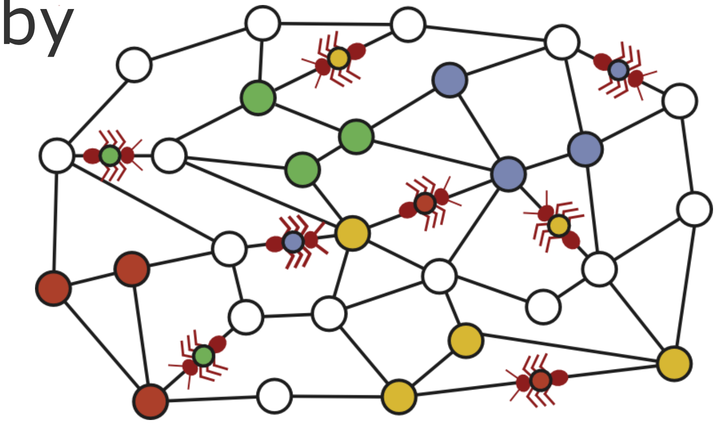
Motivation

- Scalability issues of centralized systems
- Billion triple challenge
  - DBpedia consists of 0.3 billion RDF-Statements
- Distributed systems can overcome this issues
- Few approaches on distributed RDF-storage

Flooding	**Distributed Hash Table**
Edutella (Gnutella)	RDFPeers (Chord-Ring)
	GridVine (P-Grid)

Algorithms

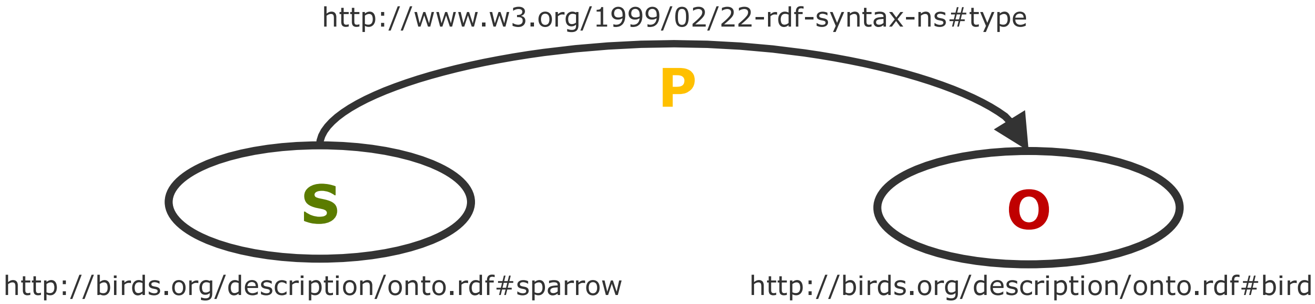
- Our approach: using swarm intelligence
- Operations are represented by ants carrying a template



- RDF-triples are considered as food respectively brood
- Pheromones for routing towards food (Foraging)
- Clustering of similar brood (Brood Sorting)
- RDF-triples are stored in three (S-,P-,O-) layers

RDF

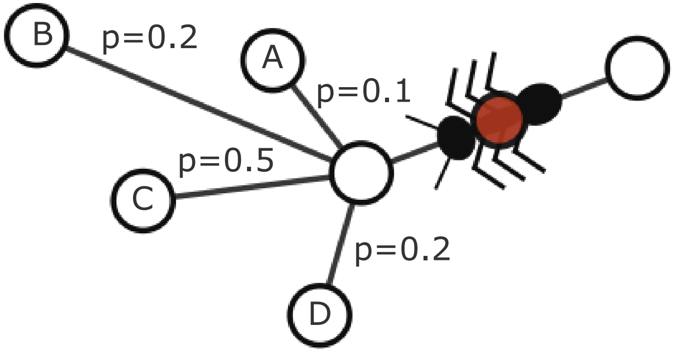
- RDF (Resource Description Framework)
  - RDF-statements (RDF-triples) are base primitive
  - Statements form a directed graph which represents knowledge
- RDF-triple <**S**ubject, **P**redicate, **O**bject>
  - Resource (**S**ubject) has a property (**P**redicate) with a value (**O**bject)
  - S,P,O have to be URIs; O can also be a literal (e.g. integer)


http://www.w3.org/1999/02/22-rdf-syntax-ns#type
P
S
O
http://birds.org/description/onto.rdf#sparrow
http://birds.org/description/onto.rdf#bird

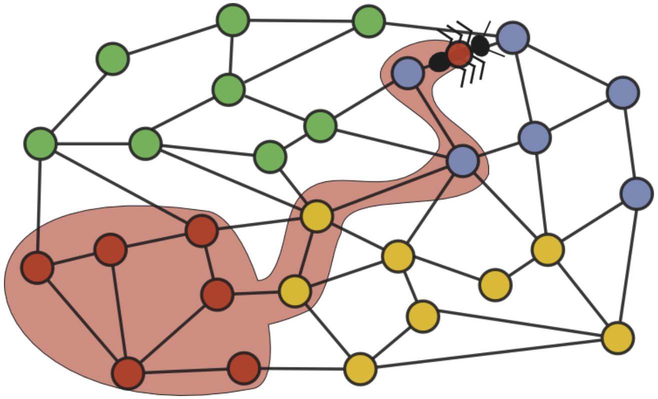
- Basic Triple Pattern e.g. <?, ?, **O**>
  - Primitive for lookup operations
  - Finds all triples with same value in the field (e.g. all triples with **O** as object)

Algorithms
Foraging

- Ants choose their next node
  - Only with local knowledge
  - At random
  - With the propability of each edge depending on the pheromone strength matching the carried template



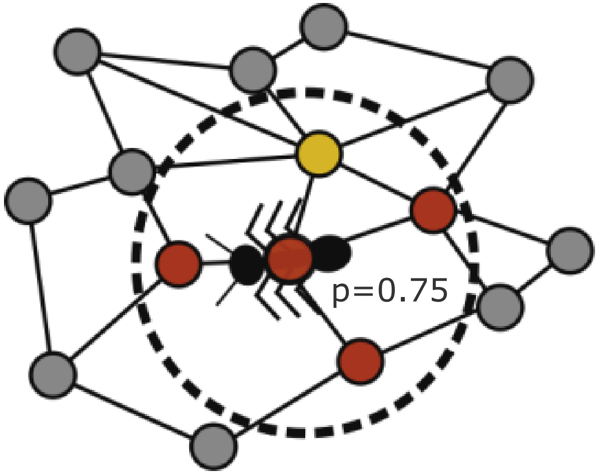
- Ants leave pheromones on their way back, which evaporate over the time



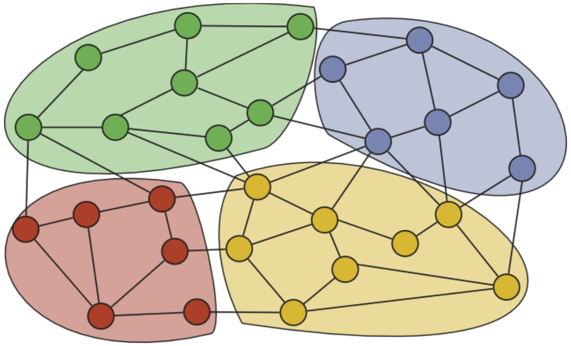
Shortest paths to clusters emerge

Algorithms
Brood Sorting (Writing)

- Data is stored on a node
  - Only with local knowledge
  - At random
  - With a propability depending on the similarity of the data to the neighbourhood
  - Only if the similarity to the current node is over a threshold value



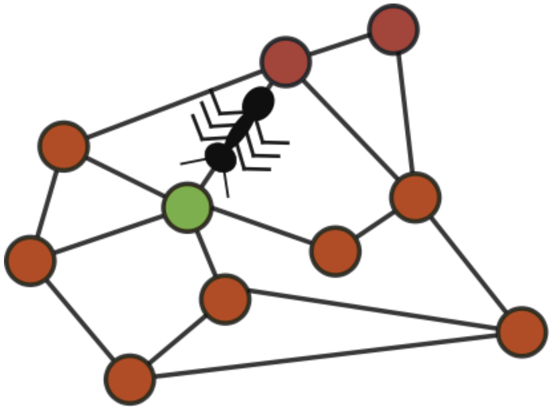
Creates clusters of similar data



- Clusters have to be maintained by moving misplaced data to its cluster

Algorithms
Brood Sorting (Cleaning)

- Misplaced data („corpses“) is lost after a certain period



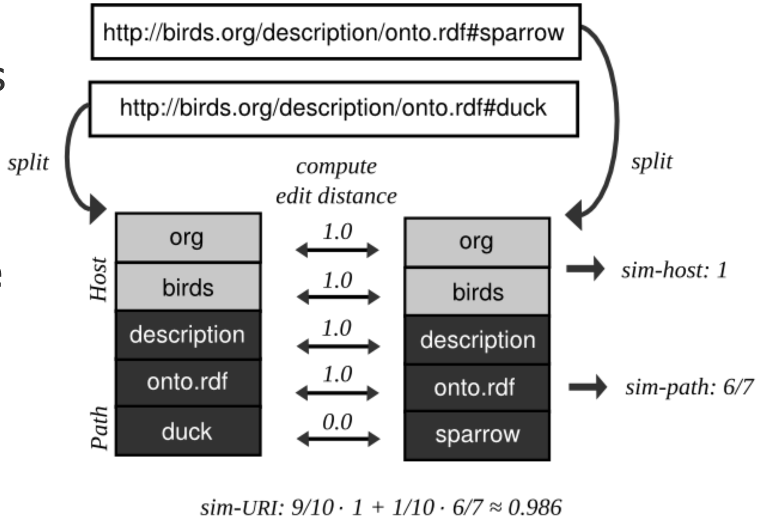
- Special ants without template roam the network randomly and clean it from corpses

- If a corpse is found, it will be moved by a spawned write-ant carrying the corpse

Leads to more homogeneous clusters und increases recall

Similarity Measures
URI Similarity

- Algorithm :
  1. Split path and domain components of the URIs
  2. Compare parts of both URIs pairwise with Levenshtein distance
  3. Sum the weighted results
  4. Similarity between the URIs is the sum



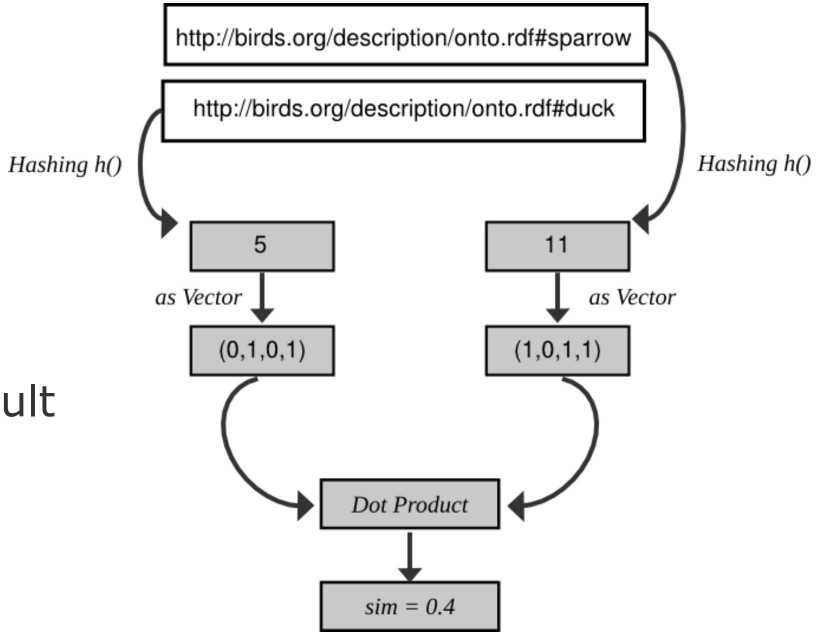
+ Supports idea that similar things having similar URIs

- Rather costly to compute

Leads to clusters following namespace scheme

Similarity Measures
Fingerprint Similarity

- Algorithm :
  1. Hash the URI Strings with h()
  2. Interpret the values as bitvectors
  3. Compare the vectors with the dot-product
  4. Similarity is the normalized result


http://birds.org/description/onto.rdf#sparrow
http://birds.org/description/onto.rdf#duck
Hashing h()
Hashing h()
5
11
as Vector
as Vector
(0,1,0,1)
(1,0,1,1)
Dot Product
sim = 0.4

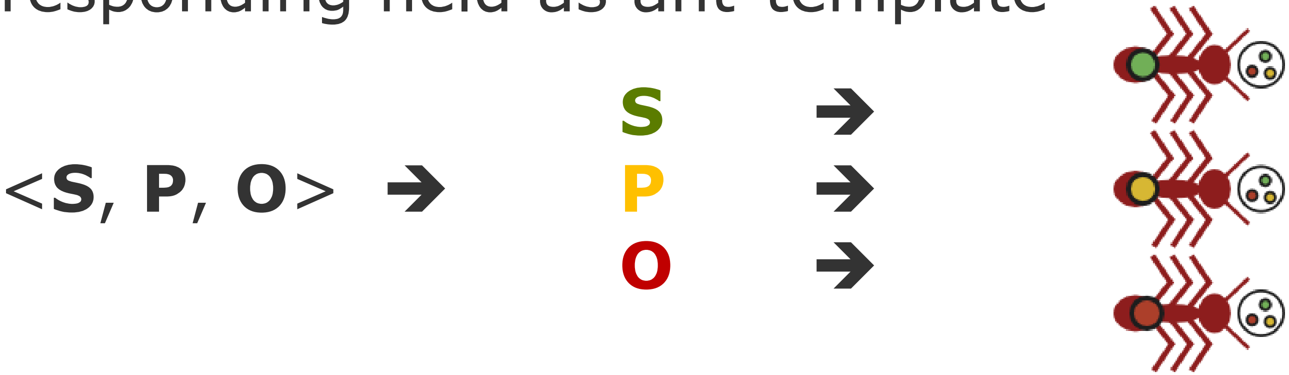
+ Very fast to compute

- Complete loss of syntactic information

Leads to uniform distributed Clusters

Write-operation

- Create three ants, one for each layer (S,P,O)
- Each ant carrying the triple and the corresponding field as ant-template



<S, P, O> →

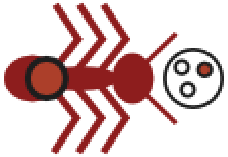
S →
P →
O →

- These ants search the cluster by following the pheromone which matches the template
- After storing the triple in its layer, the ant returns home and amplifies the pheromone

Read-operation

- Create an ant carrying the fixed part of the Basic Triple Pattern as ant-template

<?, ?, **O**> ➔ **O** ➔



- The ant follows the pheromone which matches the template and searches in the appropriate layer

- When the ant finds a result, it returns home carrying the result and amplifies the pheromone

Evaluation
Test environment

- 10 computers running 10, 20, 30, 40 nodes
  - Nodes form a random graph
  - No connected nodes on the same computer

- Test system

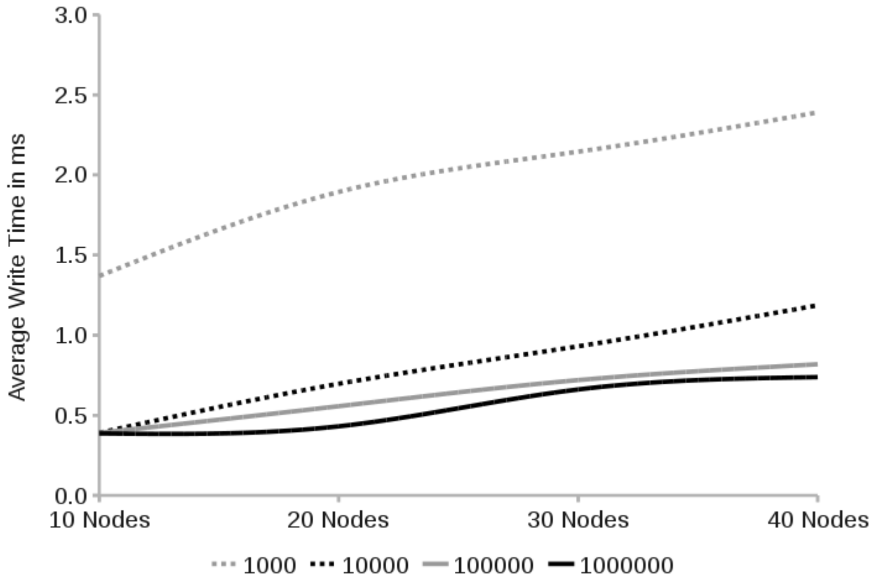
CPU	Pentium 4 2.4 Ghz
RAM	512 MByte
Network	100 MBit/s Ethernet
Operating System	Debian Linux Kernel 2.6.24
Java	1.5.0.17

- Test data

Data Source	WordNet Ontology
Write	Sets of 1K, 10K, 100K, 1M triples
Read	10k random operations for each set

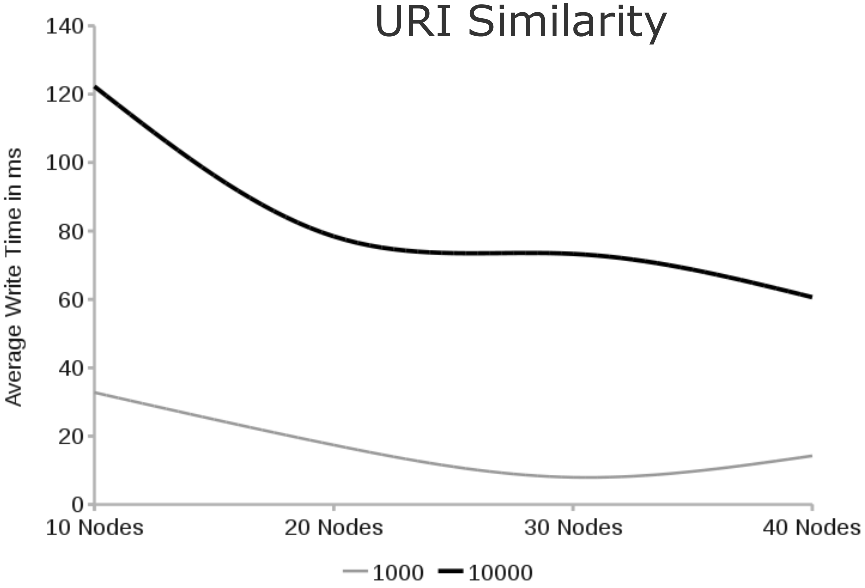
Evaluation

Write Operation



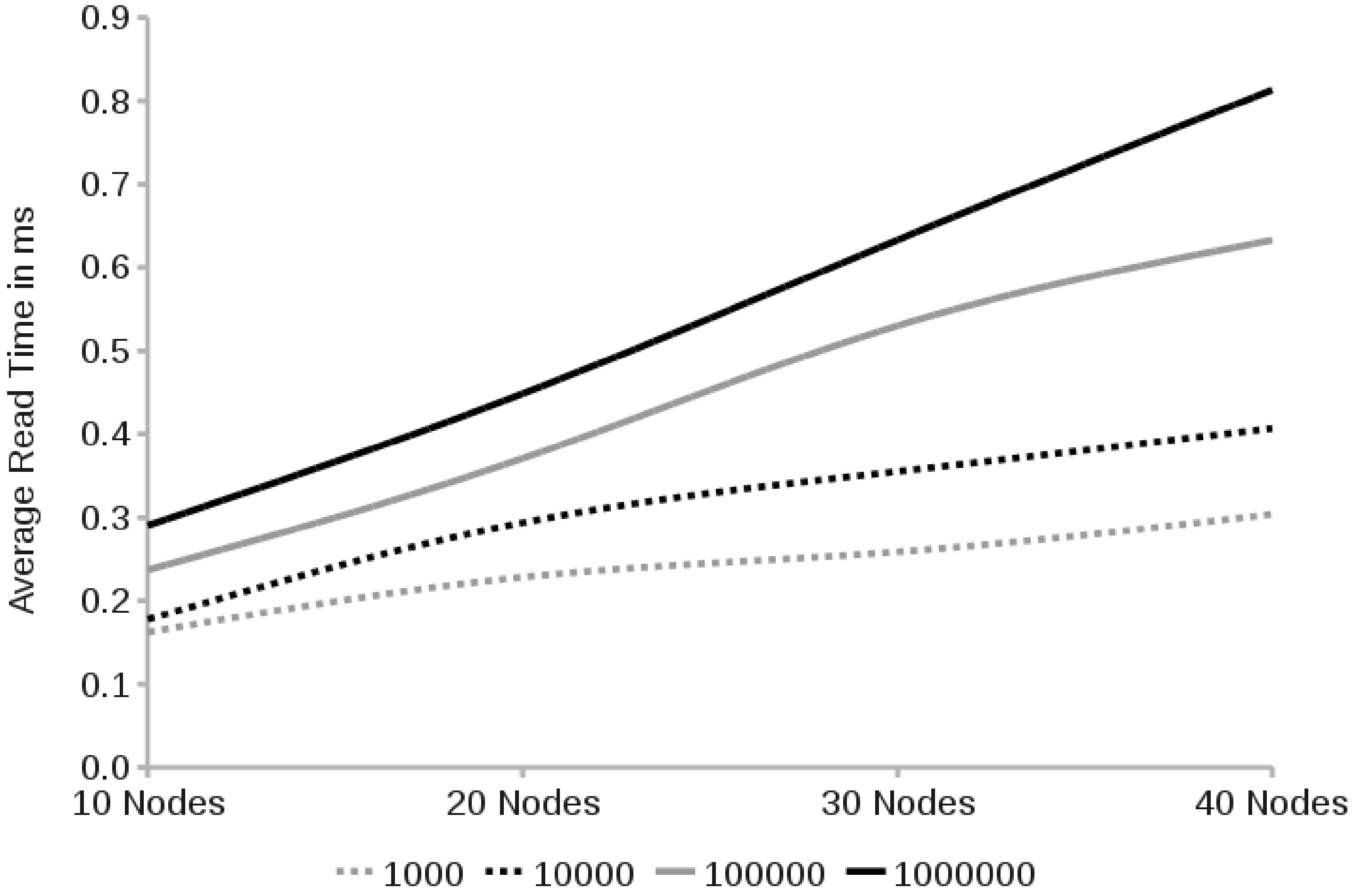
Fingerprint Similarity

URI Similarity



Evaluation

Read Operation



Conclusion

- Global knowledge is not needed for good routing and recall
- Read and write is scalable over data and nodes
- Performance of the storage layer and similarity measure has a large impact
- Fingerprinting similarity is far faster to compute, but at the cost of semantic information loss

Future work

- A fast semantic similarity (e.g. taxonomic)

- Reasoning
- SPARQL support

- Storage layer supporting similarity system

- Pheromones (ant-routing) for returning ants in place of distance-vector routing

Questions?

Thank You