Towards Swarm-based Federated Web Knowledgebases

Philipp Obermeier¹, Anne Augustin² and Robert Tolksdorf²

¹Digital Enterprise Research Institute, National University of Ireland, Galway philipp.obermeier@deri.org

²Freie Universität Berlin, Institut für Informatik, AG Netzbasierte Informationssysteme aaugusti@inf.fu-berlin.de, tolk@ag-nbi.de, http://www.ag-nbi.de

Contributions from Hannes Mühleisen² and Tilman Walther²
Modern networked applications need a scalable distributed storage infrastructure for semantic information.

Current RDF-stores are not distributed and not scalable.
out(<cat, colored, grey>)
in(<cat, colored, ?color)

The selforganized semantic storage service
S4
Data as food, query client as nest
Data as food, query client as nest
Data as food, query client as nest
Data as food, query client as nest
Data as food, query client as nest
Clusters are formed for each resource

(S,P,O)  (S,P,O)  (S,P,O)

(S,P,O)
Read:

**Require:** Template $t$, hop count $h$

1: while $h > 0$ do

2: $N = N \cup \{\text{currentNodeId}\}$

3: $T^t = \text{findMatchingTriples}(t)$

4: if not empty($T^t$) then

5: $\text{spreadScentAndReturn}(t, N)$

6: return $T^t$

7: else

8: $\text{nextNode} \leftarrow \text{selectNextNode}(t)$

9: $\text{moveTo}(\text{nextNode})$

10: $h = h - 1$

11: end if

12: end while

13: return
Write:

\textbf{Require:} Triple to store $T$, index $i$, hop count $h$, drop limit $l_d$

1: \textbf{while} $h > 0$ \textbf{do}
2: \hspace{1cm} $N = N \cup \{current\text{NodeId}\}$
3: \hspace{1cm} $p_d = calc\text{DropProbability}(T_i, h)$
4: \hspace{1cm} \textbf{if} $p_d > l_d$ \textbf{then}
5: \hspace{2cm} \textit{storeTriple}(T)
6: \hspace{2cm} \textit{spreadScentAndReturn}(T_i, N)$
7: \hspace{1cm} \textbf{else}
8: \hspace{2cm} $next\text{Node} = select\text{NextNode}(T_i)$
9: \hspace{2cm} $move\text{To}(next\text{Node})$
10: \hspace{1cm} $h = h - 1$
11: \hspace{1cm} \textbf{end if}
12: \textbf{end while}
13: \textit{storeTriple}(T)
All creatures are equal...
... but some are more equal than others

http://animals.org/onto.rdf#cat
http://animals.org/onto.rdf#dog
URI Similarity:

Take pairwise Levenshtein-distance in host and path parts eg:

$$sim_{host} = \sum_{i=1}^{\min(k,l)} c_i \cdot edit(m_{k-i}, n_{l-i})$$

Weight components along their hierarchy, eg:

$$c_i = \frac{2^{\max(k,l)-i}}{2^{\max(k,l)} - 1}$$

Weight host and path importance, eg. 0.9/0.1

Cats and dogs from animal.org are quite similar:

$$1 \cdot 0.9 + \left( \frac{2}{3} \cdot 1 + \frac{1}{3} \cdot 0 \right) \cdot 0.1 = 0.96$$
Claim: Write scales with number of nodes

Mean time to write 100K dbpedia triples over 10 test runs
Claim: Read scales with number of nodes

Median of # of hops when querying a specific triple from all nodes once
is-a relations?
in(<?animal, colored, ?color)
in(<?animal, colored, ?color)
Design principle: Only local decisions

- Local similarity measure ontology \( \times \) triple
- No global ontology

Extended behaviour of out ants:
carry triple and local type hierarchy
learn underway and merge Aboxes
determine drop probability by similarity of type carried with type dominant on node
How to implement description-logic (ontological) reasoning under the swarm intelligence paradigm?

How to integrate a reasoner on top of the swarm-based storage layer?

Idea:
Terminological axioms represented as rules
Rule applications executed by “Reasoning Ants”
Operation of a Reasoning Ant by example

T-Box Axiom:  \( \text{animal}(s) \sqsubseteq \text{flies}(s) \sqsubseteq \text{bird}(s) \)

Rule:  \( \text{bird}(s) \leftarrow \text{animal}(s), \text{flies}(s) \)
Operation of a Reasoning Ant by example

T-Box Axiom: \( \text{animal}(s) \sqsubseteq \text{flies}(s) \sqsubseteq \text{bird}(s) \)

Rule: \( \text{bird}(s) \leftarrow \text{animal}(s), \text{flies}(s) \)

1. Ant traces scents of ground instances (encoded as RDF triples) matching its rule body atoms.
2. If adequate ground facts are found the ant applies the rule: a new fact is yielded.
3. Ant stores the new fact in appropriate clusters.

\[ \text{animal}(x) \ ? \]
\[ \text{flies}(x) \ ? \]
Operation of a Reasoning Ant by example

T-Box Axiom: \( \text{animal}(s) \sqsubseteq \text{flies}(s) \sqsubseteq \text{bird}(s) \)

Rule: \( \text{bird}(s) \leftarrow \text{animal}(s), \text{flies}(s) \)

1. Ant traces scents of ground instances (encoded as RDF triples) matching its rule body atoms.
2. If adequate ground facts are found the ant applies the rule: a new fact is yielded.
3. Ant stores the new fact in appropriate clusters.
Operation of a Reasoning Ant by example

T-Box Axiom: \(\text{animal}(s) \sqsubseteq \text{flies}(s) \sqsubseteq \text{bird}(s)\)

Rule: \(\text{bird}(s) \leftarrow \text{animal}(s), \text{flies}(s)\)

1. Ant traces scents of ground instances (encoded as RDF triples) matching its rule body atoms.
2. If adequate ground facts are found the ant applies the rule: a new fact is yielded.
3. Ant stores the new fact in appropriate clusters.
Swarm-Based Reasoning Layer

*Description Logic ALC*
  for knowledge representation central core of many significant description logics

*Extended Disjunctive Rules*
  as translation target for terminological axioms
  \[
  L_1 \text{ or } \ldots \text{ or } L_k \leftarrow L_{k+1}, \ldots, L_m, \text{ not } L_{m+1}, \ldots, \text{ not } L_n
  \]
  where \( L_i \) is a literal \( A \) or \( \neg A \) for an atom \( A \)

*Partial Answer Set Semantics/Brave Reasoning*
  as model-theoretic foundation
Cluster optimization
Range queries

Multilevel indexing
Geo/Temporal

Robustness
Stability
Security
Claim: Cluster reorganization scales

Data Items vs. Move Operations random/100nodes

# of moves when inserting new triples