Towards Swarm-based Federated Web Knowledgebases

Philipp Obermeier[1], Anne Augustin[2] and Robert Tolksdorf[2]

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

[2]Freie 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 Mühleisen[2] and Tilman Walther[2]

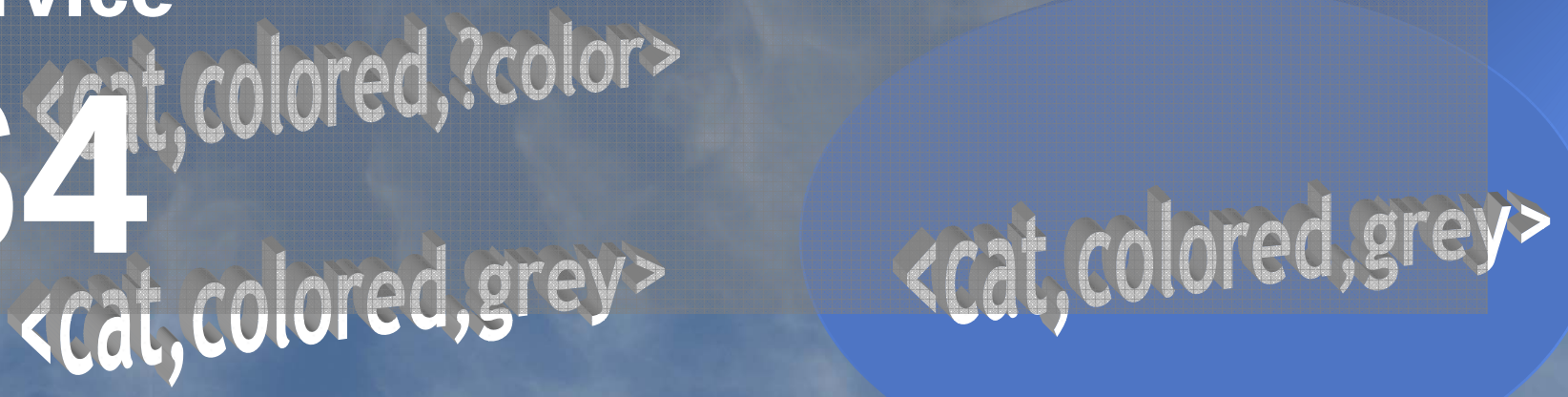
Freie Universität Berlin

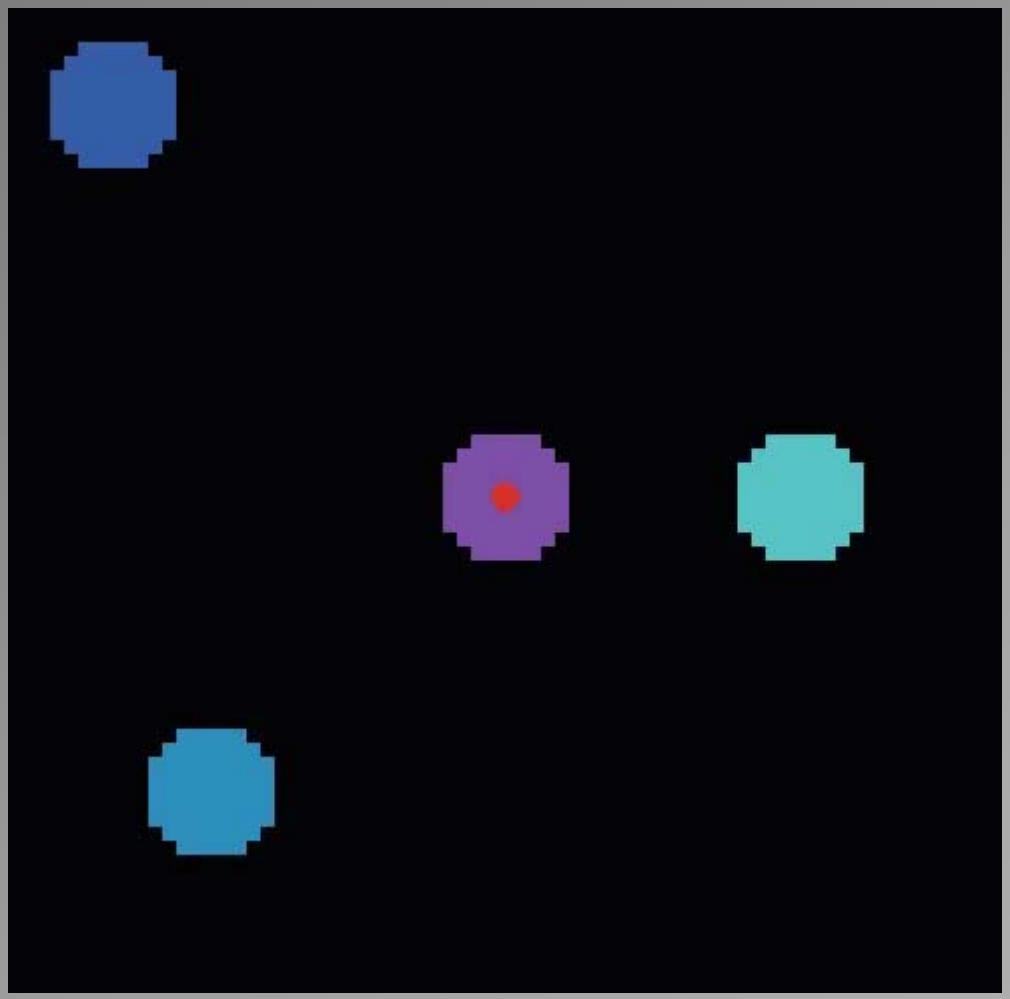

Modern networked applications need a scalable distributed storage infrastructure for semantic information
Current RDF-stores are not distributed and not scalable

out(<cat,colored,grey>)

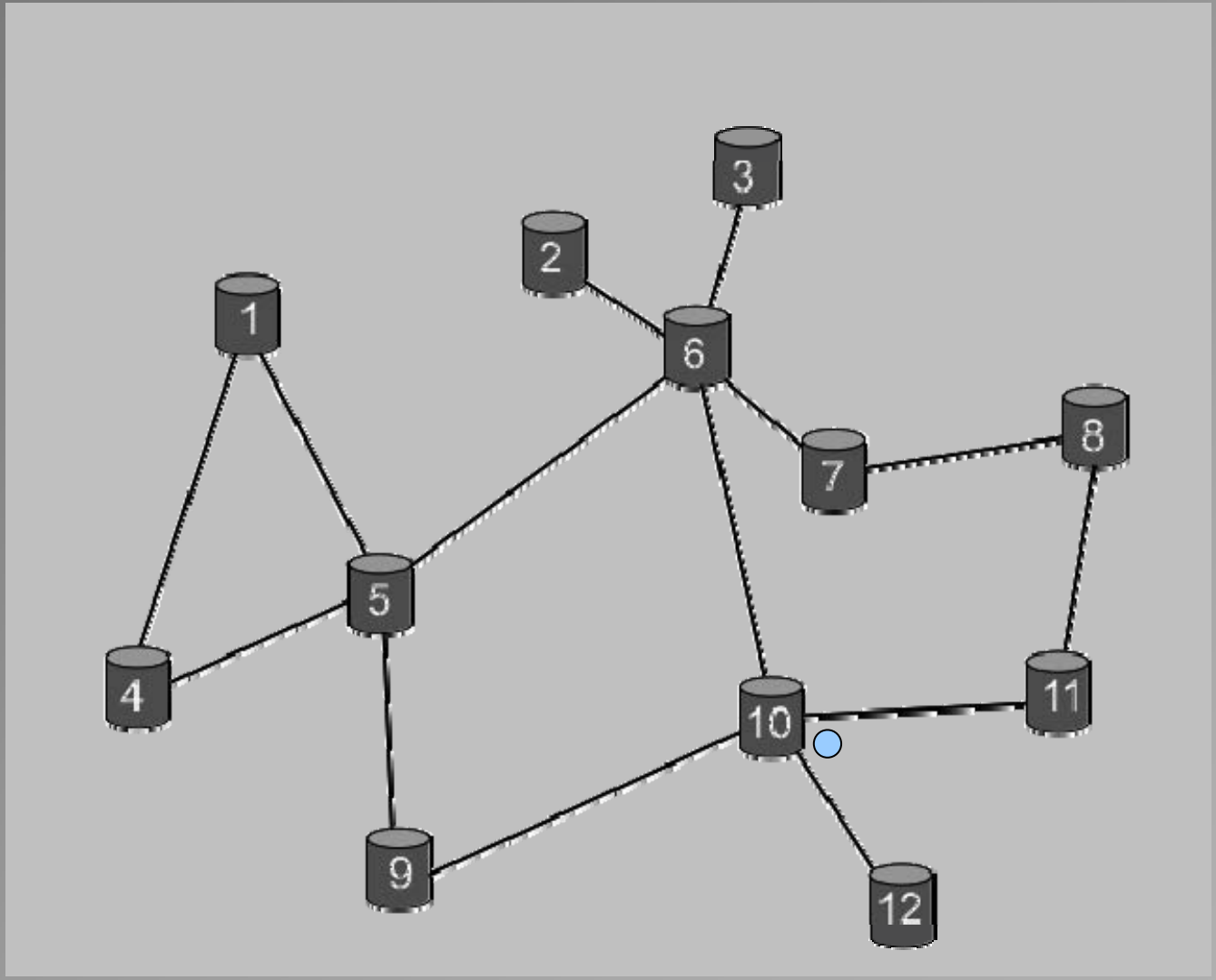
in(<cat,colored,?color)

The selforganized semantic storage service

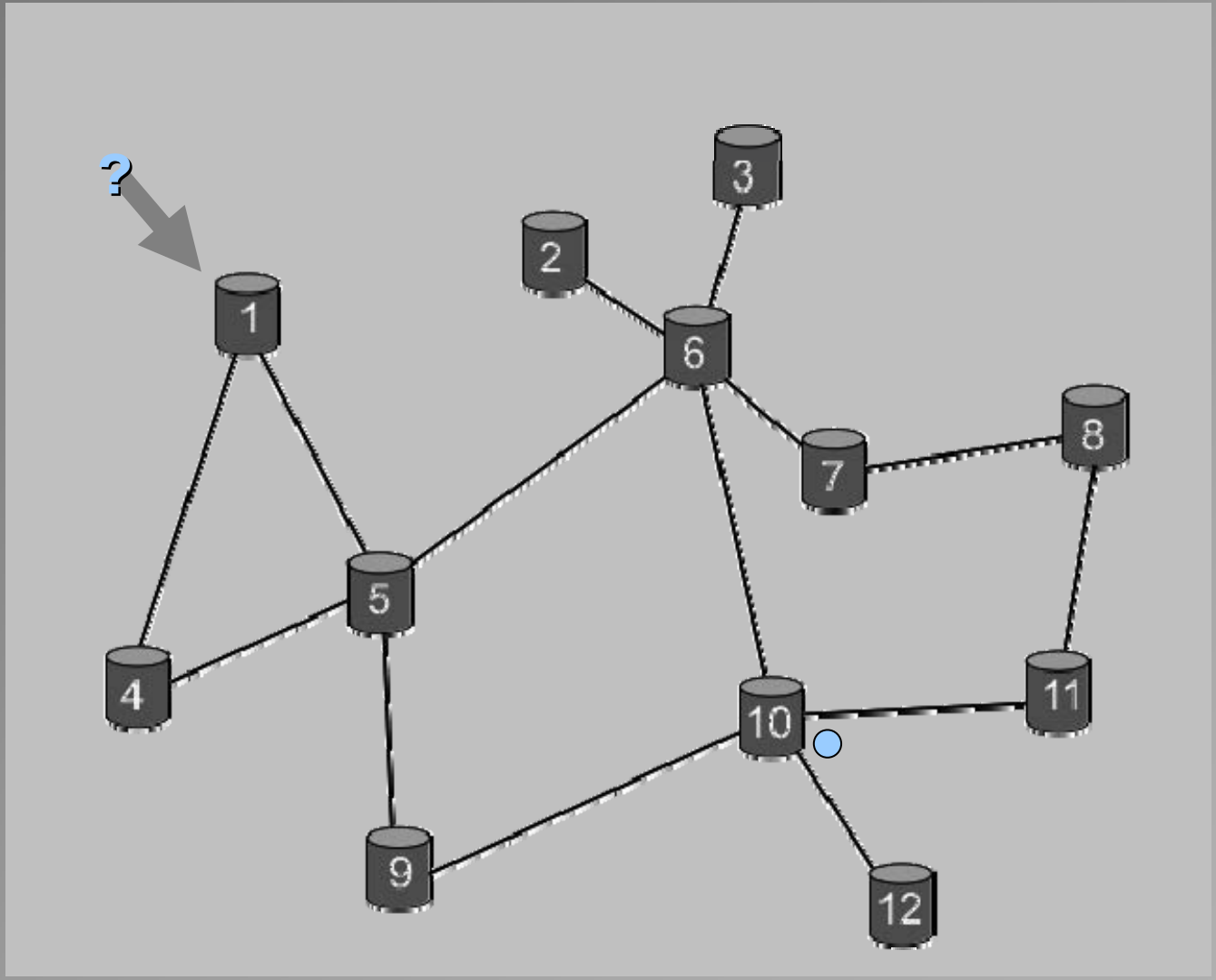




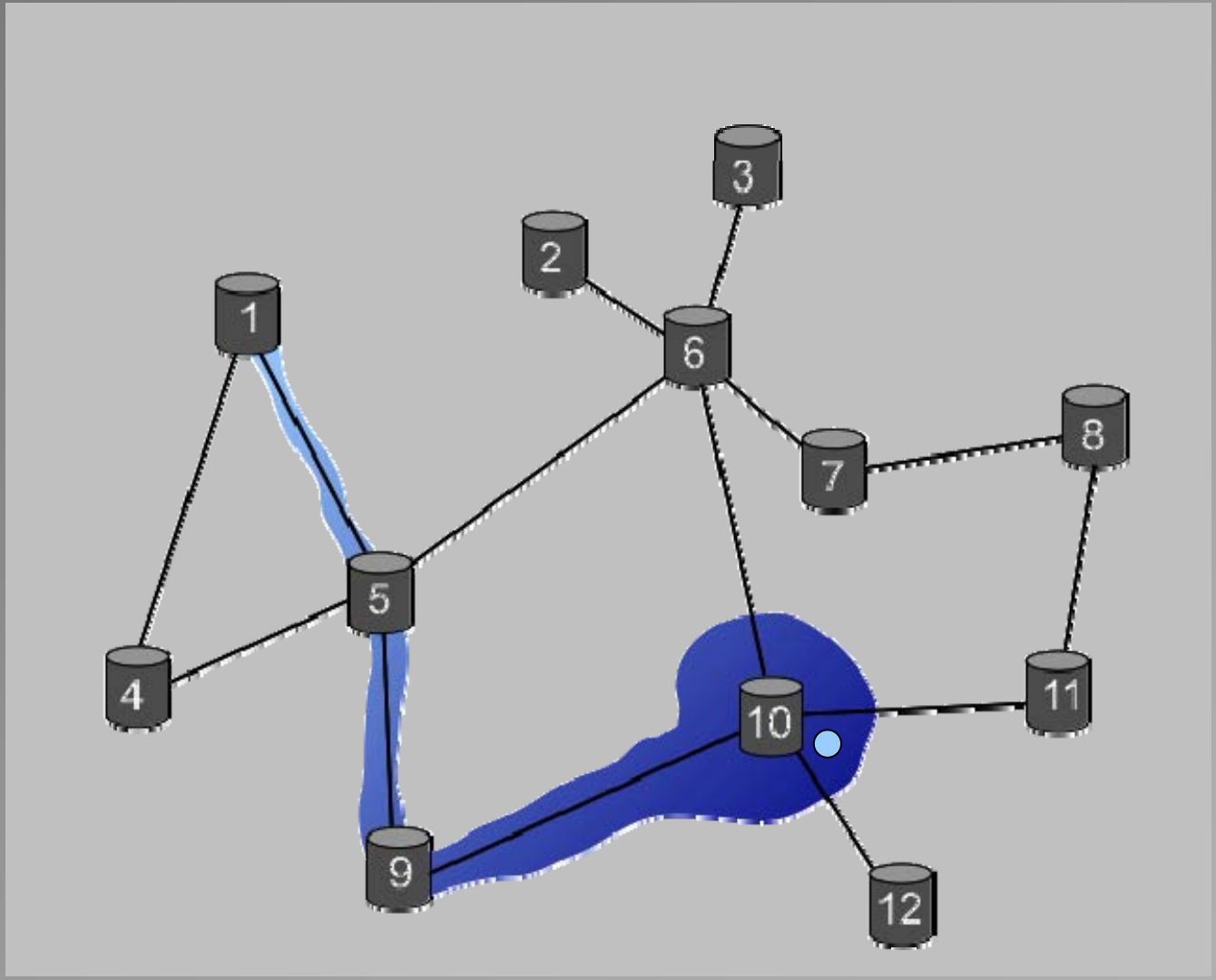
Data as food, query client as nest



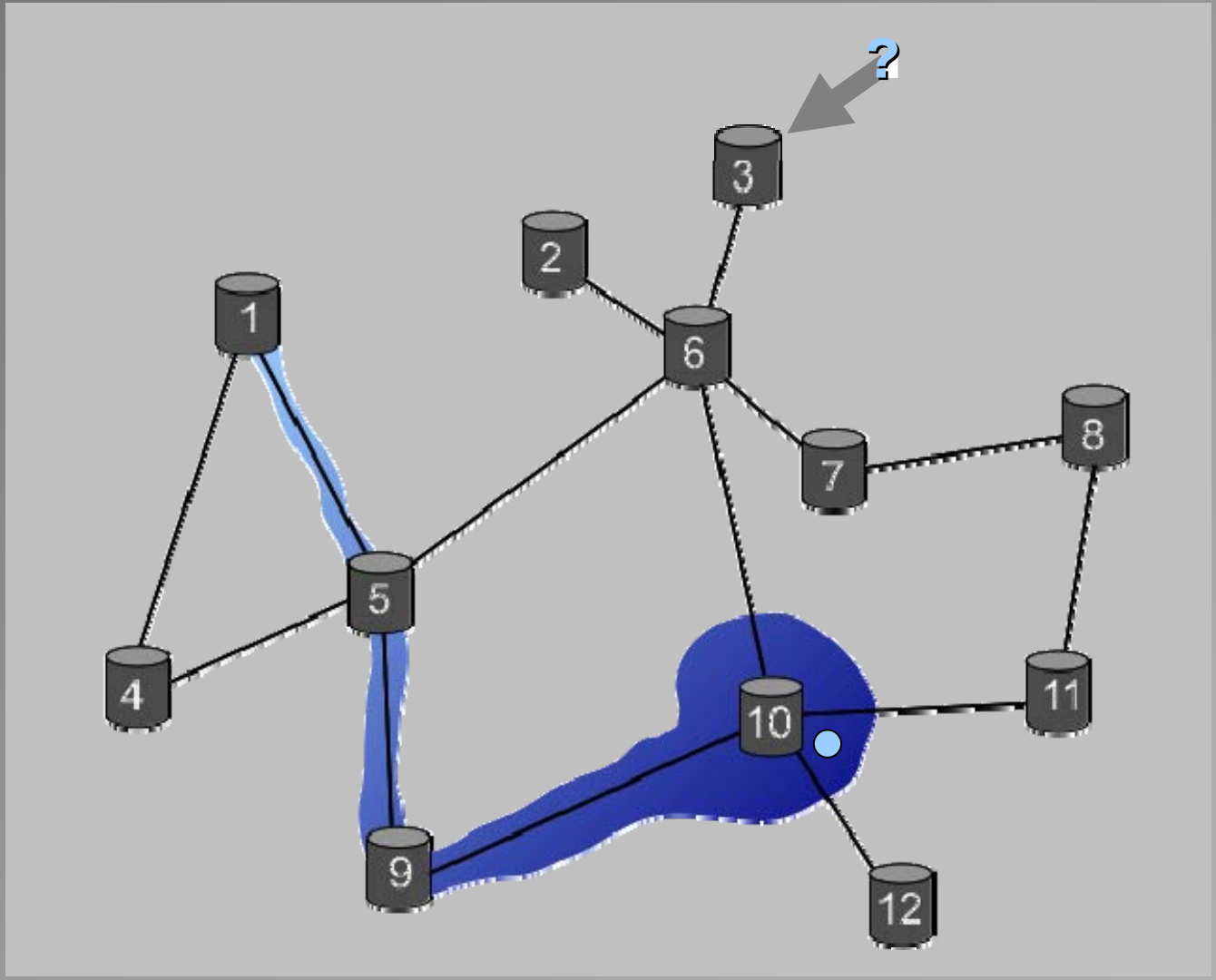
Data as food, query client as nest



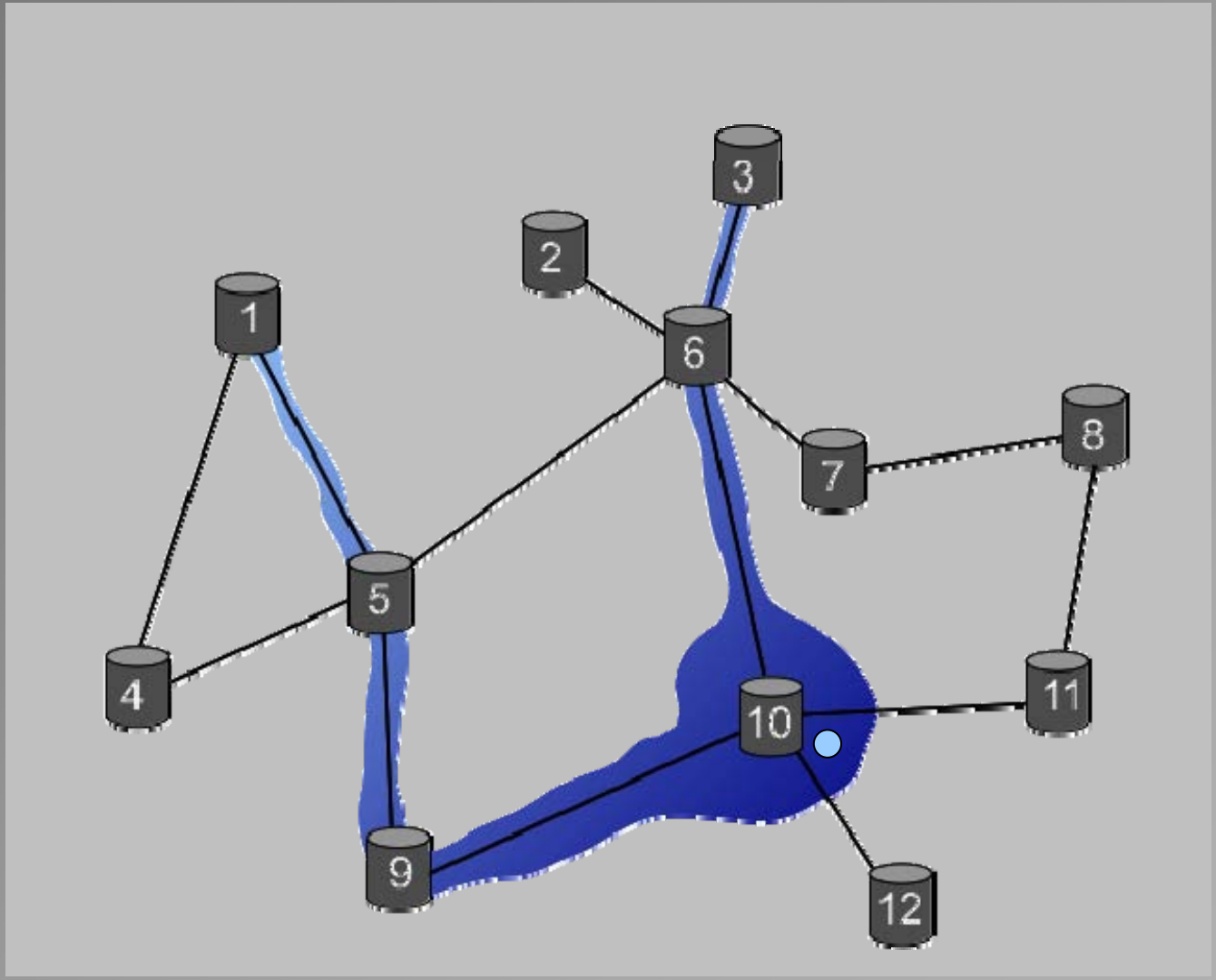
Data as food, query client as nest



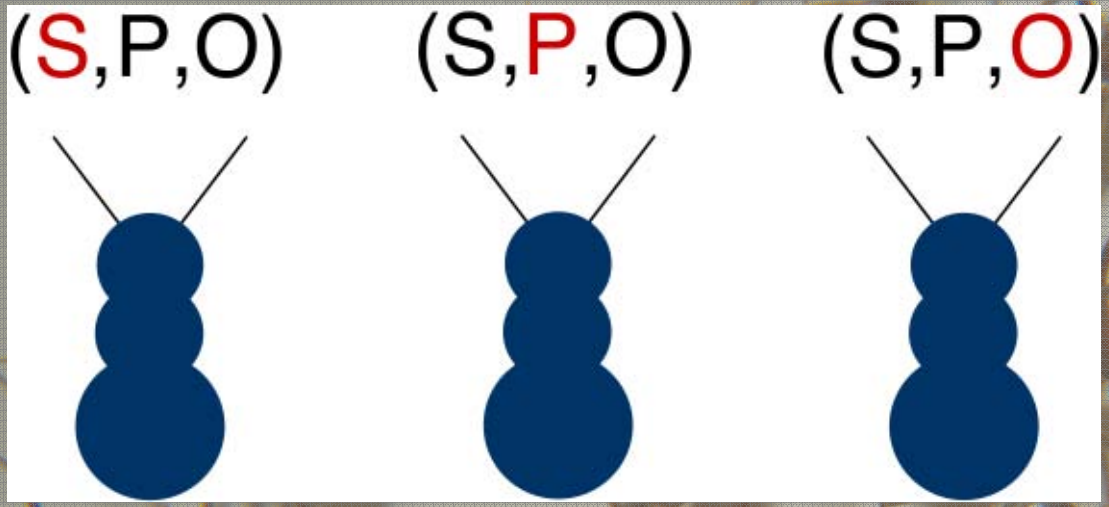
Data as food, query client as nest

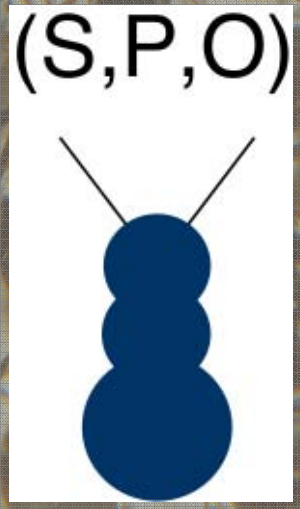


Data as food, query client as nest



Clusters are formed for each resource





Read:

Require: Template t, hop count h

1: **while** $h > 0$ **do**
2: $N = N \cup \{currentNodeId\}$
3: $T^t = findMatchingTriples(t)$
4: **if not** $empty(T^t)$ **then**
5: $spreadScentAndReturn(t, N)$
6: **return** T^t
7: **else**
8: $nextNode \leftarrow selectNextNode(t)$
9: $moveTo(nextNode)$
10: $h = h - 1$
11: **end if**
12: **end while**
13: **return**

Write:

Require: Triple to store T, index i, hop count h, drop limit l_d

1: **while** $h > 0$ **do**
2: $N = N \cup \{currentNodeId\}$
3: $p_d = calcDropProbability(T_i, h)$
4: **if** $p_d > l_d$ **then**
5: $storeTriple(T)$
6: $spreadScentAndReturn(T_i, N)$
7: **else**
8: $nextNode = selectNextNode(T_i)$
9: $moveTo(nextNode)$
10: $h = h - 1$
11: **end if**
12: **end while**
13: $storeTriple(T)$


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:

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

Weight components along their hierarchy, eg:

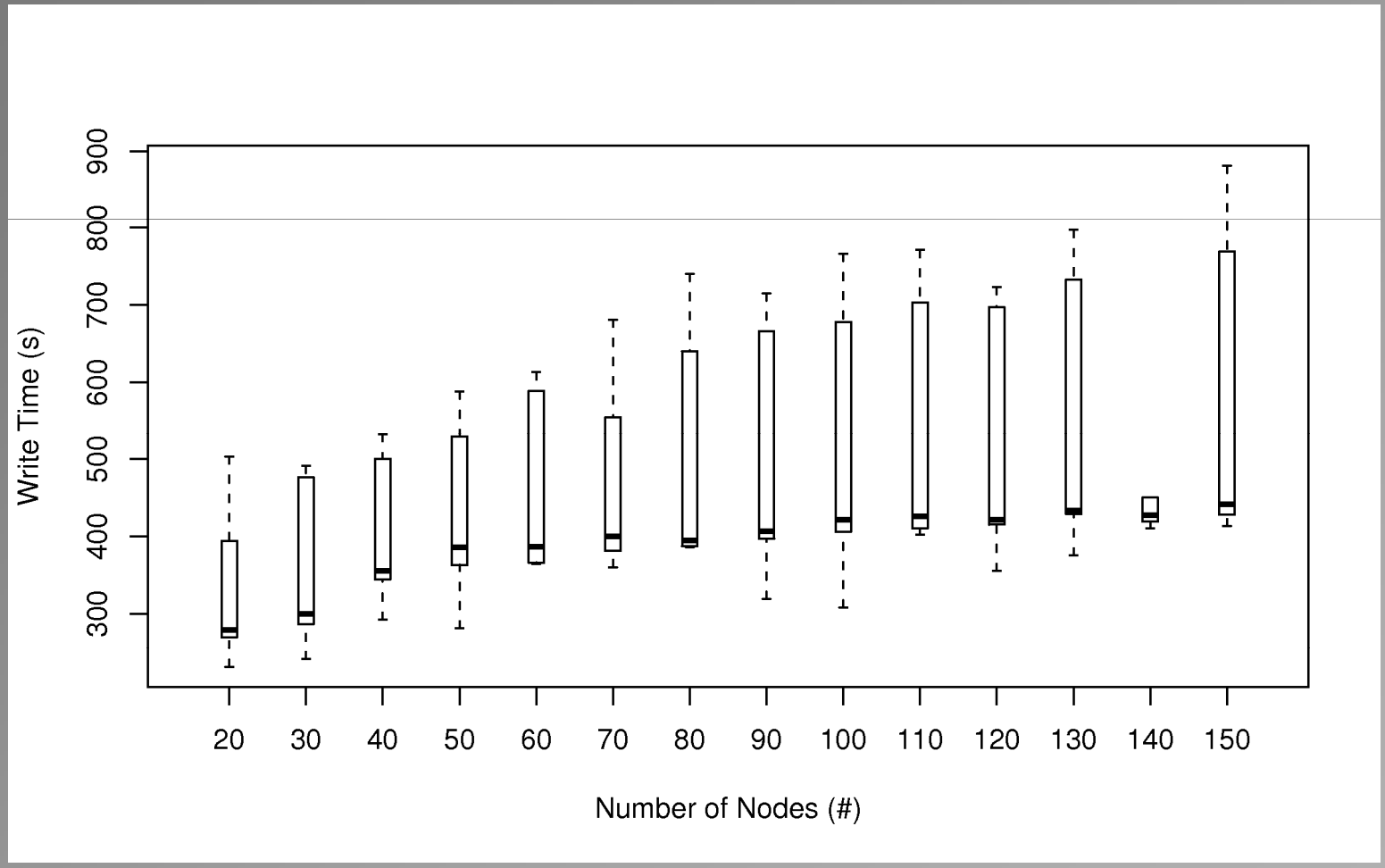
$$c_i = \frac{2^{max(k,l)-i}}{2^{max(k,l)} - 1}$$

Weight host and path importance, eg. 0.9/0.1

Cats and dogs from animal.org are quite similar:

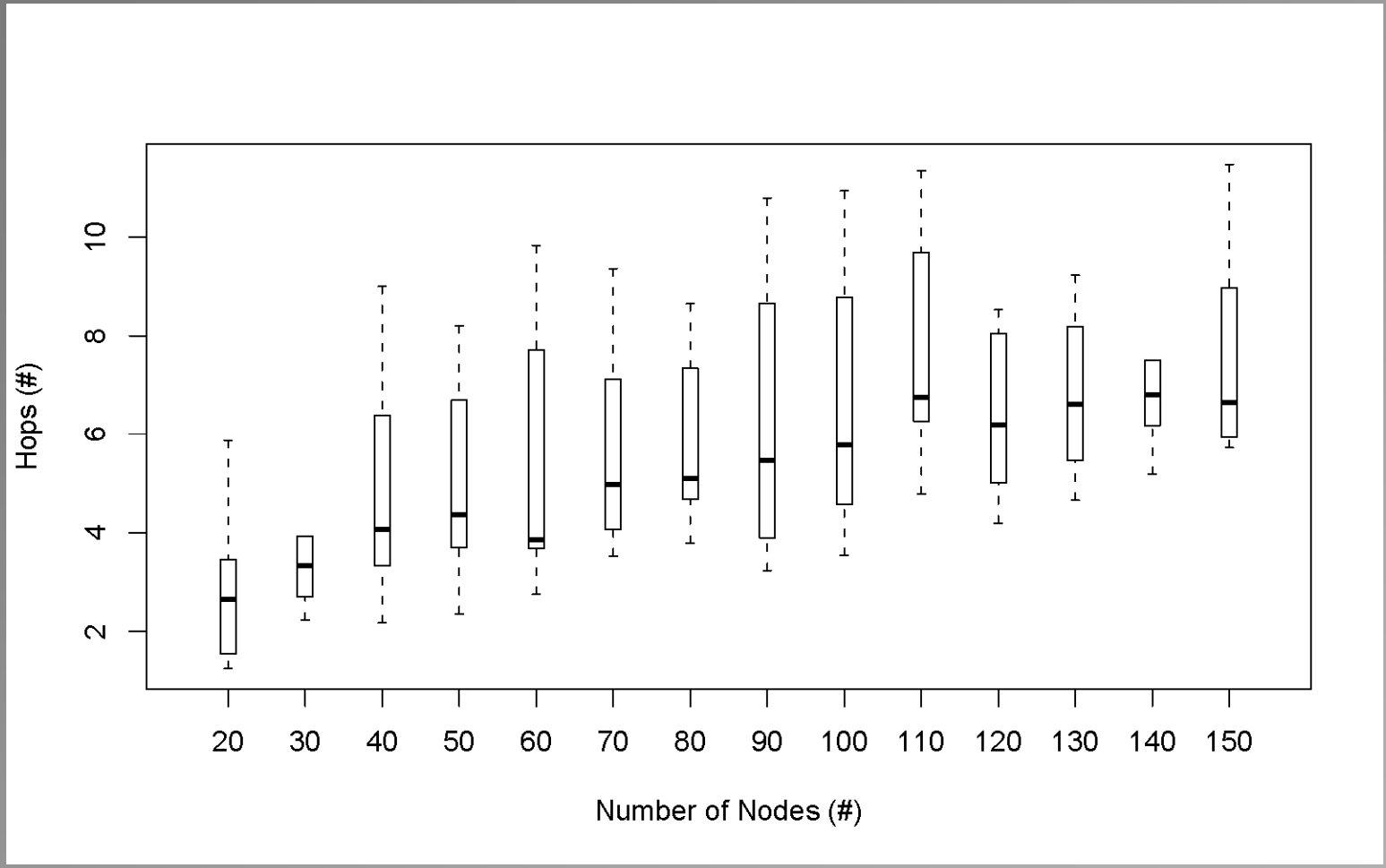
$$1 \cdot 0.9 + (\tfrac{2}{3} \cdot 1 + \tfrac{1}{3} \cdot 0) \cdot 0.1 = 0.9\overline{6}$$

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 quering a specific triple from all nodes once

is-a relations?

in(<?animal,colored,?color)

in(<?animal,colored,?color)

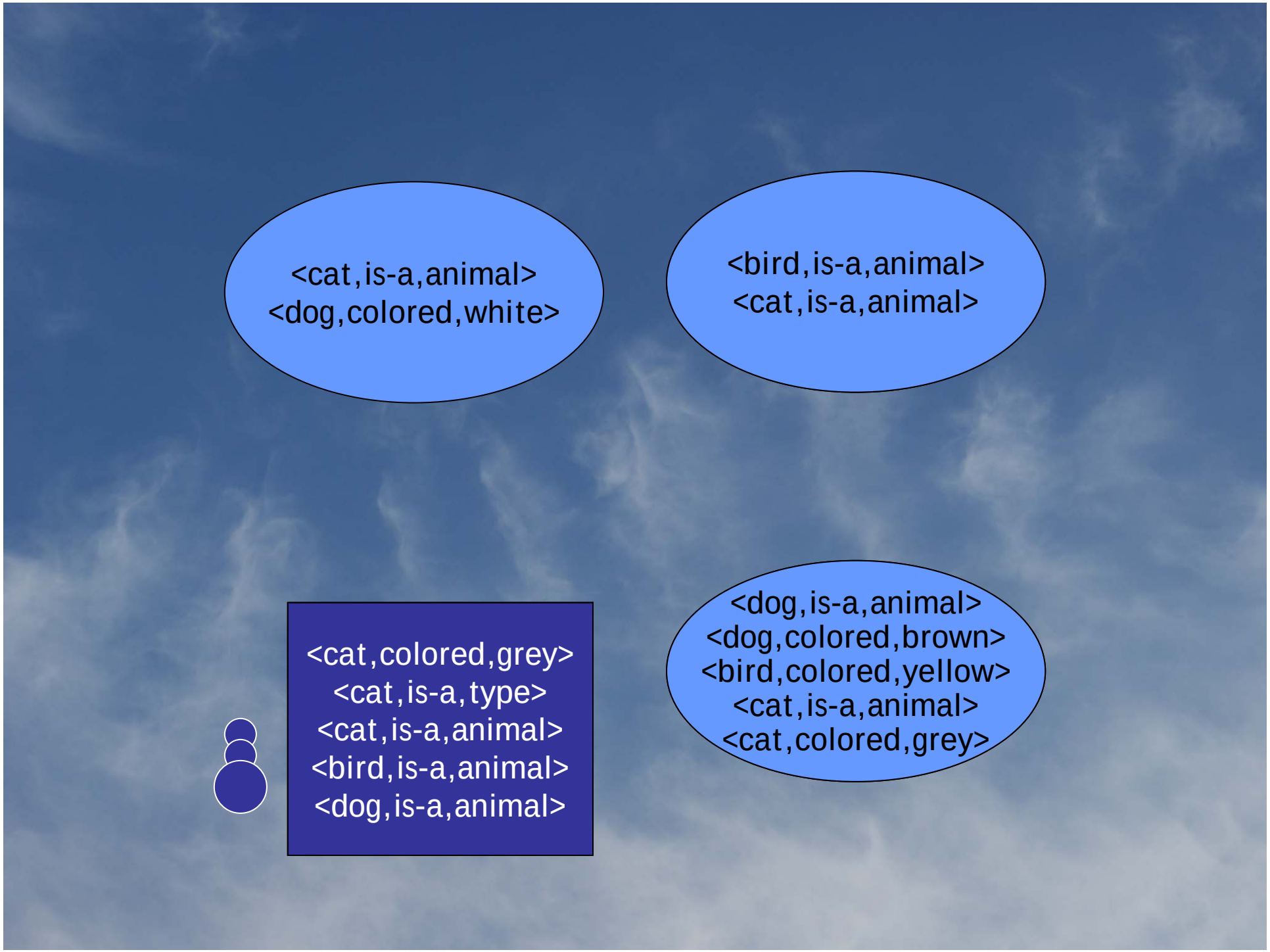
Design principle: Only local decisions

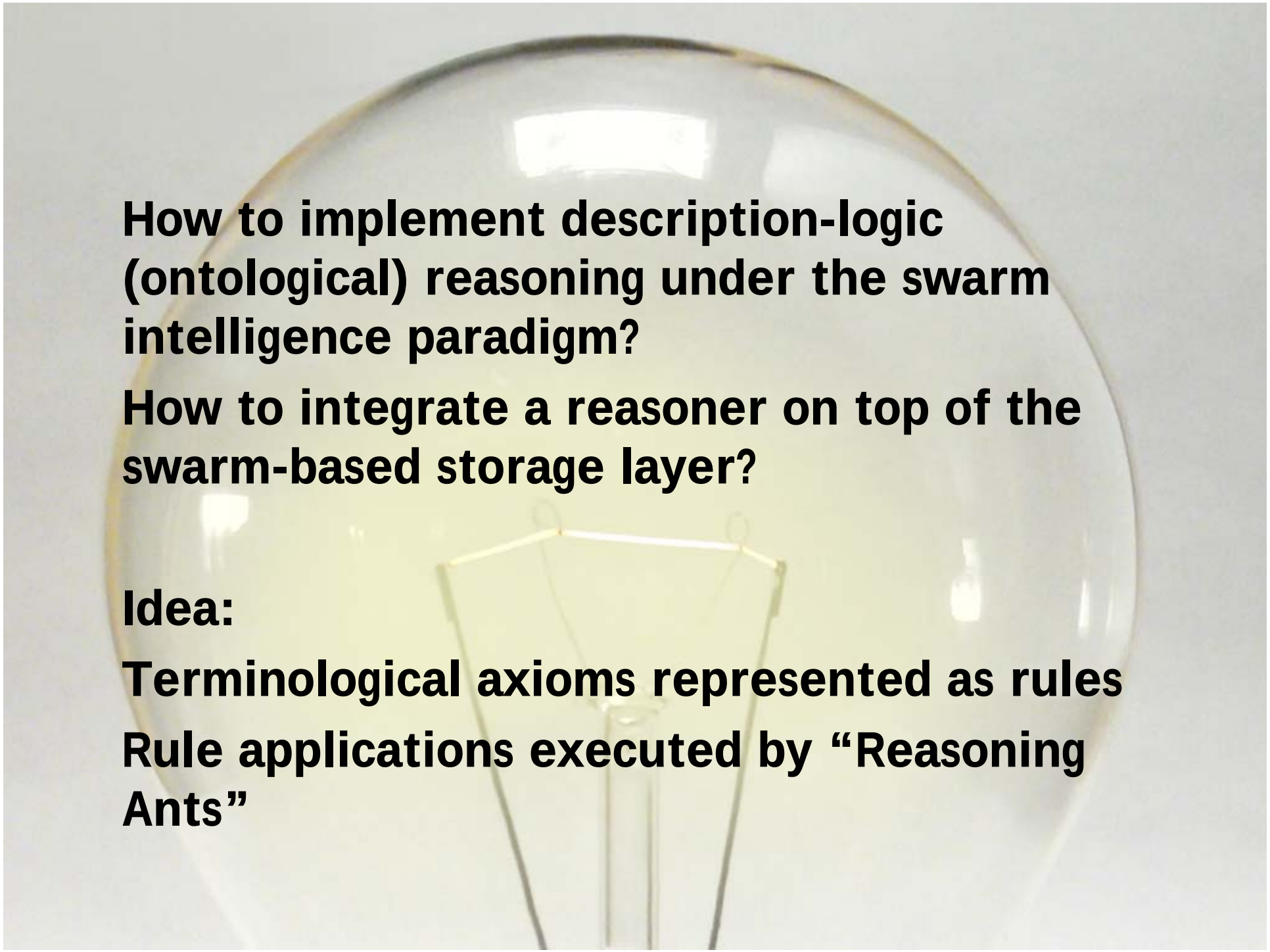
⇨ Local similarity measure ontology x triple

⇨ 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


<cat,is-a,animal>
<dog,colored,white>
<bird,is-a,animal>
<cat,is-a,animal>
<cat,colored,grey>
<cat,is-a,type>
<cat,is-a,animal>
<bird,is-a,animal>
<dog,is-a,animal>
<dog,is-a,animal>
<dog,colored,brown>
<bird,colored,yellow>
<cat,is-a,animal>
<cat,colored,grey>



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:

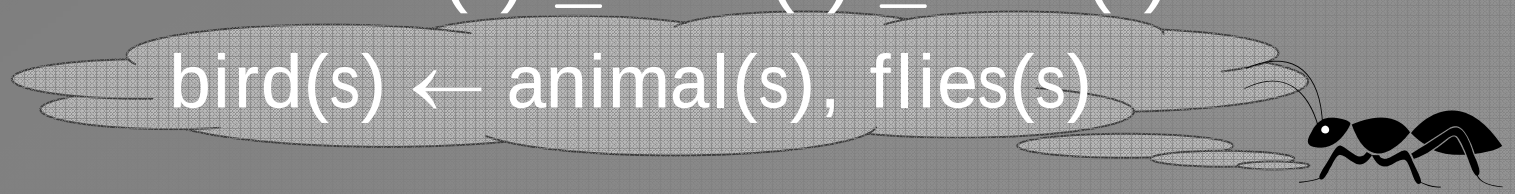
Terminological axioms represented as rules

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

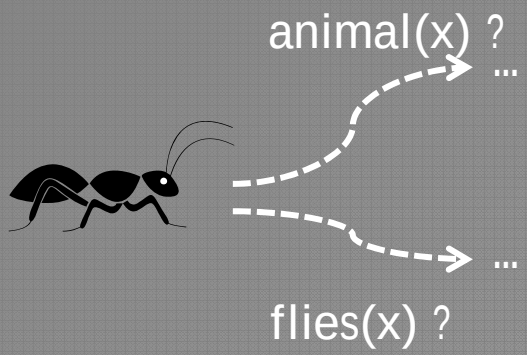


Operation of a Reasoning Ant by example

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.

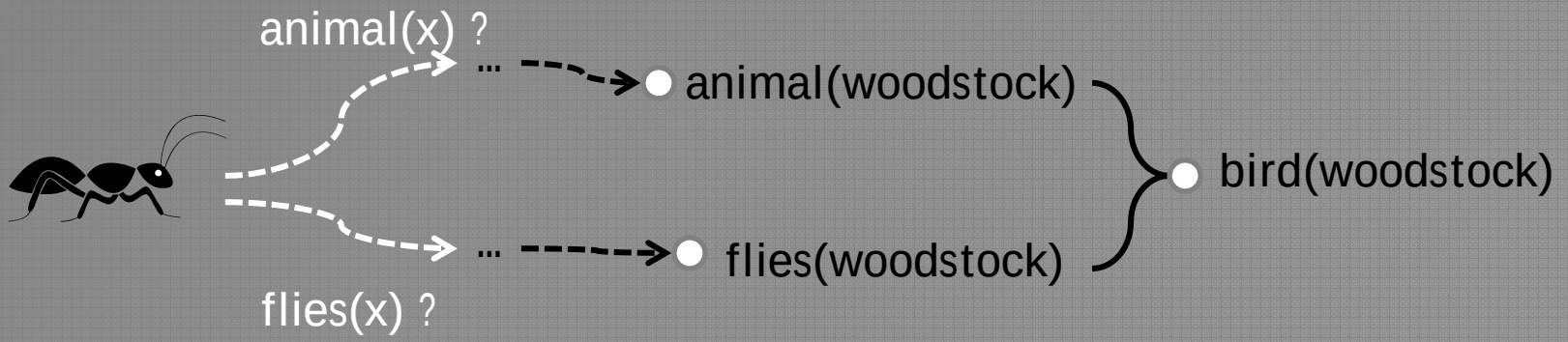


Operation of a Reasoning Ant by example

T-Box Axiom: $animal(s) \sqsubseteq flies(s) \sqsubseteq bird(s)$

Rule: $bird(s) \leftarrow animal(s), flies(s)$

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: animal(s) $\sqsubseteq$ flies(s) $\sqsubseteq$ bird(s)

Rule: bird(s) $\leftarrow$ animal(s), 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 or … or $L_k \leftarrow L_{k+1},\ldots, L_m$, not $L_{m+1},\ldots,$ not L_n

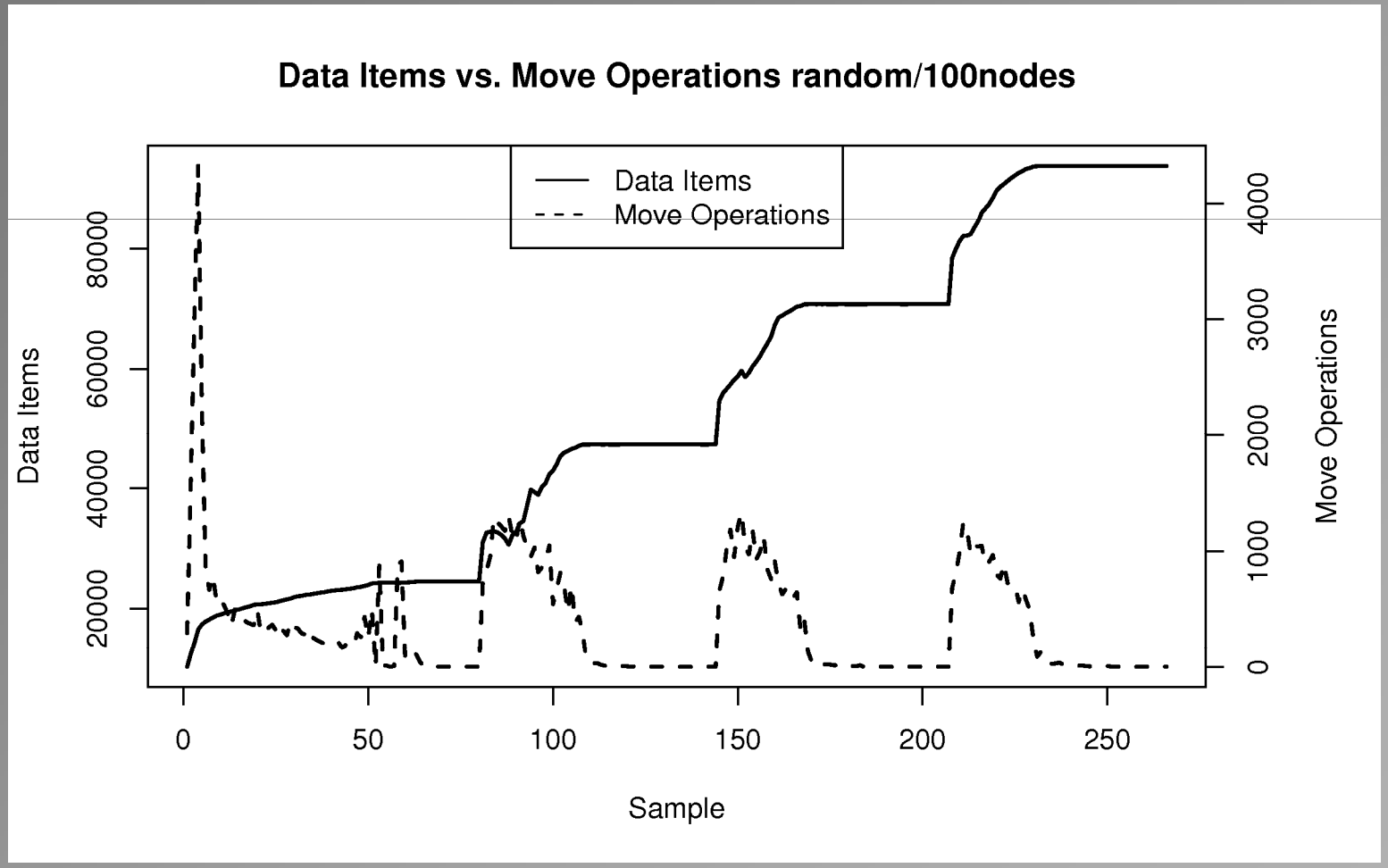
where L_i is a literal A or $\neg$A for an atom A

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



# of moves when inserting new triples