# A Swarm-based Semantic Storage Service

Hannes Mühleisen, Kia Teymourian and Robert Tolksdorf

Networked Information Systems Group
Department of Computer Science - Freie Universität Berlin
{muehleis, kia}@inf.fu-berlin.de, tolk@ag-nbi.de
http://www.ag-nbi.de

**Abstract.** The amount of data handled by semantic applications is expected to increase over a level manageable by available storage systems. Distributed semantic storage solutions are a promising way to increase storage capacity, but current approaches often rely on static network structures. We are in the process of developing a Self-Organized Semantic Storage Service using swarm intelligence to overcome the limitations in storage capacity and network dynamics.

## 1 Introduction

Most Semantic Web applications require a semantic storage service. These storage services have evolved from standalone systems to distributed solutions running on multiple computers (here referred to as *nodes*). Distributed solutions have to achieve a compromise between *scalability* and *dynamicity*. Previous systems either rely on special nodes orchestrating request processing with focus on scalability, or maintain a Peer-to-Peer overlay network structure of equal nodes achieving a compromise, or simply use communication techniques such as flooding with their focus on dynamicity. An optimal system should be able to scale to an arbitrary amount of nodes and triples and also be able to tolerate network changes without overhead.

Our group has been working on swarm-based distributed tuple storage for several years [1]. These concepts have been extended to handle RDF triples [2]. A main advantage over P2P systems maintaining a global network structure is the ability to scale beyond the limits of those architectures. Swarm-based storage systems for semantic data have the potential to store and query semantic data over a large number of nodes, thus being able to store huge amounts of RDF data. In our current research project DigiPolis, we are in the process of developing such a system. We refer to this system as the Self-Organized Semantic Storage Service (S4) and present the main ideas and algorithms of our storage service here.

## 2 The Self-Organized Semantic Storage Service (S4)

A S4 storage system is comprised of a network of nodes running identical software. Each node is stores a share of the data set stored in the entire network. Nodes keep track of a limited neighborhood of other nodes, but have no knowledge of the entire network. User client programs can connect to any node and issue basic operations like write and read. Requests are then routed to the nodes where the affected data is stored.

S4 is based on ant behavior patterns, thus basic operations are translated into "virtual ants". Ants move between the nodes, they perform all tasks using strictly local knowledge. Clustering of related information is achieved on the basis of a similarity measure linking semantically related information. This enables the system to scale well by performing all operations in a very limited part of the network, thus effectively outperforming previous approaches where data is distributed without insight.

For each triple to be stored, virtual ants are spawned and moved through the network using the respective nodes neighborhood. On each node visited, they determine whether a triple is supposed to be stored on the current node. This calculation includes the amount of similar triples stored on this node, the current system load, the amount of hops left and a random factor. If a triple is stored, the ant returns to the node it originated from. On the way back, a "scent" for the triple just stored is spread, in order to enable efficient triple retrieval as described below.

After a read request has been issued to an arbitrary network node, a number of virtual ants is spawned to retrieve the triples requested by a triple pattern, for example (ex:foo, ?p, ?o) for triples with "ex:foo" as an explicit subject. The ant starts searching for matching triples on the current node. If no triples are found, it selects and moves to another node from the neighborhood using the scents present. If triples are found, the ant returns them to the node the read request was issued to.

## 3 Conclusion

We have already commenced our work on the design and implementation of S4 components, and first results show our system to be as scalable as expected. Various algorithms have been shown to perform very well in simulations and prototypical implementations. However, evolving the algorithms simulated to production versions and tuning the various configuration parameters for our self-organizing system requires further extensive evaluation. We will implement several alternatives, define the corresponding configuration sets, and both simulate and test the assembled systems on a number of network configurations, data distributions, and load scenarios. Our implementation of S4 will form the base for an indoor navigation system able to cover the buildings for an entire city. We are planning to extend S4 with scalable reasoning capabilities to support inferencing on huge amounts of semantic data.

## Acknowledgments

This work has been supported by the "DigiPolis" project funded by the German Federal Ministry of Education and Research (BMBF) under grant number 03WKP07B.

## References

1. Ronaldo Menezes and Robert Tolksdorf. A new approach to scalable linda-systems based on swarms. In *Proceedings of ACM SAC 2003*, pages 375–379, 2003.
2. Robert Tolksdorf and Anne Augustin. Selforganisation in a storage for semantic information. *Journal of Software*, 4, 2009.