DISTRIBUTED ALLOCATION OF A CORPORATE SEMANTIC WEB

Ana B. Rios-Alvarado, Ricardo Marcelín-Jiménez and R. Carolina Medina-Ramírez
Department of Electrical Engineering, Universidad Autónoma Metropolitana-Iztapalapa, Atlixco186, DF, Mexico

Keywords: Semantic Web, Ontologies, Distributed Storage.

Abstract: This paper outlines a general method for allocating a document collection over a distributed storage system. Documents are organized to make up a corporate semantic web featured by a graph G_1, each of whose nodes represents a set of documents having a common range of semantic indices. There exists a second graph G_2 modelling the distributed storage system. Our approach consists of embedding G_1 into G_2, under space restrictions. We use a meta-heuristic called "Ant Colony Optimization", to solve the corresponding instances of the graph embedding problem, which is known to be a NP problem. Our solution provides an efficient mechanism for information storage and retrieval.

1 INTRODUCTION

The semantic web is an extension of the current web intended to provide an improved cooperation between humans and machines (Berners-Lee, 2001). This approach relies on a formal framework where information is given a well-defined meaning. It uses ontologies to support information exchange and search. It is also based on semantic annotations to code content representation. In addition, it works with formal knowledge representation languages in order to describe these very ontologies and annotations.

There exist diverse contexts that may profit from the ideas developed by the semantic web. Corporate memories, for instance, share common features with the web, both gather heterogeneous resources and distributed information and have a common concern about the relevance of information retrieval. Nevertheless, corporate memories have an infrastructure and scope limited to the organization where they are applied. Among the heterogeneous resources belonging to a scientific group or enterprise, for example, documents represent a significant source of collective expertise, requiring an efficient management including storage, handling, querying and propagation. From this meeting between the web and the corporate memories a new solution is born, the Corporate Semantic Web (CSW). Formally, a CSW is a collection of resources (either documents or humans) described using semantic annotations (on the document contents or the persons features/competences), which rely on a given ontology (Gandon, 2002).

This paper describes a general method to allocate a CSW in a distributed storage system. A system such as this is a collection of interconnected storage devices that contribute with their individual capacities to create an extended system offering improved features. The importance of this emerging technology has been underlined in recent research works. Although its simplest function is to spread a collection of files across the storage devices attached to a network, desirable attributes of quality must also be incorporated.

Our proposal, that we call a semantic layer, is mainly defined by its two main procedures: information location and query. Location solves document placement and creates the tables supporting query. We consider this approach provides a flexible, scalable and fault-tolerant service. Location is solved using a meta-heuristic that accepts either a centralized or distributed implementation. We believe this may lead to a self-configurable semantic storage system.

The remaining of this paper is organized as follows. Section 2 is an overview of the previous work which is related to our proposal. Section 3 presents the general structure of our semantic layer. Section 4 is a formal statement of the Graph Embedding problem. Section 5 is about our simulation method. Section 6 describes the

Rios-Alvarado A., Marcelín-Jiménez R. and Medina-Ramírez R. (2009).
DISTRIBUTED ALLOCATION OF A CORPORATE SEMANTIC WEB.
In *Proceedings of the International Conference on Knowledge Management and Information Sharing*, pages 173-179
DOI: 10.5220/0002302501730179

experimental agenda and results. Section 7 closes our paper with a summary of our contribution and directions for further work.

2 RELATED WORK

Semantic storage on P2P systems is a very active area of research. Different proposals have been developed in order to address the many aspects of this issue. Risson and Moors focused on information retrieval (Risson and Moors, 2006), based on the utilisation of semantic indices. They devised two alternative mechanisms called VSM (Vector Space Model) and LSI (Latent Semantic Index). In VSM, any document or query is featured by a vector of terms or indices. Therefore, a query defines a point in the corresponding vector space. In contrast, LSI performs a keyword query not only on the indices that feature the documents, but also on the contents. This way it is possible to find documents having a semantic proximity with the query, although they may not have any index matching the initial quest. PeerSearch (Tang, 2002) is a project that included both mechanisms to support document indexing.

In (Wolf-Tilo, 2005) we found an information recovery service based on the classification of contents in a digital library, on a P2P system too. There exist projects like (Kjetil, 2006) and (Crespo and Garcia-Molina, 2002) advocating the utilization of semantic overlays on top of P2P storage networks, already deployed. In SOWES (Kjetil, 2006), for instance, peers are gathered in neighbourhoods or zones, based on keyword vectors describing their contents. Next, each zone collects vectors from its corresponding nodes and rebuilds new clusters based on vector similarities. Also, (Crespo and Garcia-Molina, 2002) propose a semantic overlay network (SON) on top of a music storage system. This approach allows queries to be forwarded to the corresponding SON. A SON creation is based on the hierarchical classification of those concepts describing the semantic profile of potential documents. Then, files are stored under their corresponding SON. Retrieval starts with a user query which is classified, i.e. linked with a key concept within the hierarchy.

Heterogeneity is another important matter. Although there might be many P2P storage systems supporting semantic retrieval, it does not mean that exchange is possible between any two of them. Besides its own resources and the description of the underlying knowledge domain, each local system must adopt a “trade” convention in order to be a part of an exchange agreement. In a Peer Data Management Systems (PDMS), for instance, each member has a mapping mechanism that translates any external query into its local scheme. With this approach, the extended system is able to render documents to any query, from any reachable place. Piazza (Halevy, 2003), Edutella (Nejdl, 2002), and RDFPeers (Cai, 2004) are extended systems, built on these principles.

Finally, ANTHILL (Montresor, 2001), is a framework providing support for P2P applications development. It is based on a biological model: ants foraging behaviour. It considers low level features such as communications, security and scheduling. As for document storage, ant algorithms follow a draconian policy where those files frequently used, are preferred over those seldom consulted.

3 SYSTEM DESCRIPTION

It is a well-known fact that the IP routing task, at the Internet, is supported by two complementary procedures: the first one, which performs table keeping and the second one, which performs table querying. Following similar principles, we propose the storage of a given CSW based on two procedures. First, we solve document location and build a table, whose entries show the places in charge of a given set of documents. Second, we perform look-up on this table in order to consult the corresponding contents.

This paper addresses our methodology for document location. From our view, a given CSW is a set of documents classified according to an underlying ontology. This ontology itself describes a collection of concepts (categories) which can be mapped to an ordered set of indices. Each document is therefore labelled with an index. Let us assume that the CSW is partitioned in such a way that, all the documents sharing a common index are associated to the same subset. Now, the CSW can be modelled by means of a graph G_1, each of whose nodes represents a subset of the partition. Every documental node $v_i \in G_1$ is featured by 2 parameters, the range $r_{i1}...r_{i2}$ of semantic indices spanning its corresponding documents and the weight $\omega(v_i)$, which is the information these documents amount to.

Figure 1 is an example of a given CSW, once it has been prepared for storage. There exist 17 different concepts (categories) which, in turn, define 17 indices. Nevertheless, the whole set of documents it is encompassed by a graph having only 5 nodes,

each of them featured by its range and weight. Two nodes sharing a link are assumed to have related concepts.

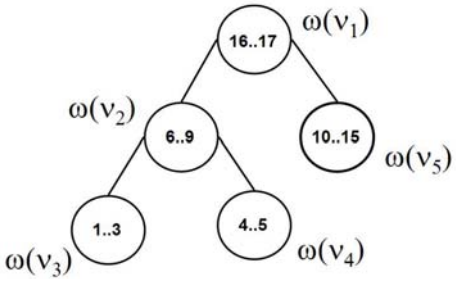

Figure 1: CSW modelled by G_1.

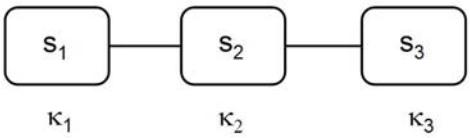

Figure 2: Storage network modelled by G_2.

In the complementary part of our description, we model the storage network using a second graph G_2. Each node $v_j \in G_2$, from now on store, has an associated capacity $\kappa(v_j)$ that features the maximal amount of information it is able to contain. Fig. 2 is an example of storage network. Each store shows its capacity. We say that the storage network has homogeneous capacity if, for any store v_j, $\kappa(v_j) = k$.

Document location implies the embedding of G_1 into G_2. This problem consists of using as few stores as possible in order to place as many documental nodes as possible inside these stores, in such a way that their aggregated weight does not exceed the corresponding capacity. When the particular instance of graph embedding is solved, each store receives a copy of the look-up table. Each row in this table has two parts; the left entry indicates a semantic indices range, while the right entry indicates the store in charge of the documents in this range. Fig. 3 shows how G_1 has been embedded into G_2. Based on this solution we have built table 1.

Any user looking for some content in the CSW may contact any store and ask for a given index. Using its local table, the store will be able to recognize whether or not it keeps the corresponding documents. In the first case, it immediately turns in the documents to its client. Otherwise, it works on behalf of its client and contact the proper store to retrieve the documents matching the user's query.

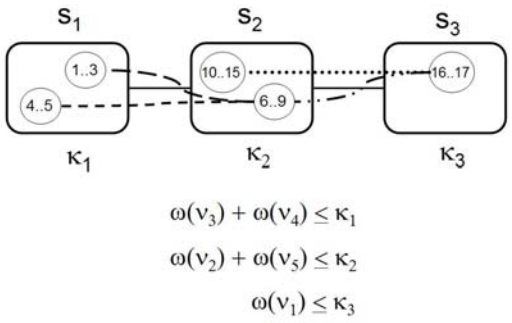

Figure 3: The embedding of G_1 into G_2.

Table 1: Look-up table.

Indices range	Store
1..5	s_1
6..15	s_2
16..17	s_3

The down side of graph embedding is that it is an NP-complete problem. Nevertheless, there exist a vast collection of meta-heuristics developed to tackle this family of problems within bounded accuracy and reasonable complexities. We decided to address our instances of graph embedding using the ant colony optimization method (ACO). It is a probabilistic technique for solving hard computational problems. ACO is inspired by the behaviour of ants in finding efficient paths from their anthill to the places where they collect their food. ACO dates back from the pioneering work of Dorigo (Dorigo, 1992), but it was a few years ago when scientists started regarding this technique as a very promising candidate to develop distributed meta-heuristics, due to its cooperative and inherently distributed nature.

4 PROBLEM STATEMENT

Let $G_1 : (V_1, E_1)$ be a graph representing a CSW and let $G_2 : (V_2, E_2)$ be the graph representing a storage network. The embedding of G_1 into G_2 is a couple of maps $S : (\nu, \varepsilon)$, where ν assigns each node from G_1 to a store in G_2, while ε transforms paths in G_1 to paths in G_2, upon the following restriction:

There exists a function $\omega : V_1 \mapsto R$, called weight. Also, there is a function $\kappa : V_2 \mapsto R$, called capacity. Let $N_j = \{v_i \,|\, v_i \in V_1, v_j \in V_2, \nu(v_i) = v_j\}$ be the set of nodes from V_1 mapped to $v_j \in V_2$.

$$\sum_{v_i \in N_j} \omega(v_i) \leq \kappa(v_j), \forall v_j \in V_2$$

$$\bigcup_j N_j = V_1$$

$$N_j \bigcap N_{j'} = \varnothing, j \neq j'$$

This is, the total weight of nodes in V_1 stored in $v_j \in V_2$, must not exceed the corresponding capacity. Consequently, our optimization problem can be defined.

Problem (GE). Find an embedding S of G_1 into G_2 such that, for a given function $f: S \mapsto R$ that measures the cost of a given embedding, S has the lowest cost $f(S)$.

For the goals of this work, we will assume that G_2 has homogeneous capacity, this means that each store has the same capacity k. Also, we will assume that there is a linear ordering L on the stores in G_2, such that $succ(v_j)$ is the successor of $v_j \in V_2$, according to L, or null if v_j is the last element in L. GE is an NP-complete problem (Savage, 1991).

5 ANT COLONY OPTIMIZATION

Even though it is accepted that the initial work in ACO was due to Dorigo, we will adopt a description by Gutjhar (Gutjahr, 1999), which is slightly different, but fits better with our exposition.

Our implementation consists of creating Z scout ants. Every ant is charged to perform a random depth first search on G_1. As each ant travels across the graph, it associates the visited nodes to a given store $v_j \in G_2$. When the aggregated nodes' weight exceeds the capacity of the current store, it reassigns the last node to $succ(v_j)$ and starts this filling process over again, provided that there are still nodes to visit. Every ant exhaustively visits G_1 and reports its solution path to the common nest. We call this procedure traversal. Depth first search (DFS) is the building block of this stage. It is implemented using a well known distributed algorithm (Cidon, 1988) with time and message complexities $O(n)$ and $O(m)$, respectively. Where n is the order of G_1, and m its size.

In due time, the nest analyzes the cost of each reported solution. Then, it spreads more or less pheromone on each path, depending on its quality, i.e. according to a defined evaluation function; a good path receives more pheromone than a bad one. Prizing, as it is also called this stage, is carried out using propagation of information with feedback (PIF) (Segall, 1983). It is a distributed algorithm with time and message complexities $O(D)$ and $O(n)$, respectively. Where D and n are the diameter and the order of G_1, respectively. As a by-product, it also builds a spanning tree on G_1. From this moment on, our method profits from this spanning tree, in order to perform successive prizing.

Then, the nest starts the next cycle with Z new scouts that will perform the same procedure: traversal. Nevertheless, even though it still is a random search, the prizing phase biases the new resulting paths. Rounds are repeated for a fixed number or until the difference between the best solution of two consecutive rounds does not exceed a given bound.

6 EXPERIMENTS AND ANALYSIS

Once we had a running simulator, we designed a set of experiments in order to evaluate the quality of our method and its complexity. Let us recall that, as we work with pseudo-random number generators, each individual simulation requires a new seed to grant a new collection of results. So, from now on, when we describe a single experiment, we mean the same simulation repeated under 10 different seeds.

From our point of view, we consider that an instance of the problem is solved when 75% of the agents follow the same trail, which represents the best solution, i.e. the least number of stores from G_2 able to allocate all the nodes from G_1.

In the first set of experiments we investigated the initial number of ants Z that produces, during the first round, the highest variability on the resulting trails. The explanation is that we wanted to bind the resources required to produce the biggest amount of independent random trails (potential solutions) on G_1. For nodes in G_2, we fixed their storage capacity c=500. Next, for nodes in G_1, we designed three different initial graphs, with 100, 300 and 600 nodes, respectively. In turn, each of them produced 5 new graphs, with common order but random weights uniformly distributed between 0 and 500. Then, we tested each of the 15 resulting graphs under 7 different values of Z, 5, 10, 15, 20, 25, 30 and 35. This means that we devised 105 experiments.

According to the different instances of the problem, and the 7 levels of agents that we tried,

Table 2: Variance for the number of stores in the first round.

	$n = 100$			$n = 300$			$n = 600$		
Ants(Z)	μ	σ^2	σ	μ	σ^2	σ	μ	σ^2	σ
5	60.3	1.63	1.27	158.3	1.32	1.14	308.2	1.43	1.19
10	56.4	0.95	0.97	160.1	2.31	1.52	301.3	3.79	1.94
15	58.1	1.23	1.11	158.9	2.55	1.59	308.6	3.65	1.91
20	59.2	0.89	0.94	157.2	2.40	1.55	304.4	3.82	1.95
25	51.4	0.45	0.67	158.3	1.63	1.27	301.2	5.34	2.31
30	55.5	0.56	0.74	154.4	1.20	1.09	305.1	5.45	2.33
35	59.3	0.50	0.70	156.5	1.26	1.12	307.2	5.36	2.32

table 2 shows the mean(μ), variance(σ^2) and standard deviation(σ) for the number of stores required to contain a given number of documental nodes. This table, as well as, fig. 4, suggests that there is a minimum number of agents producing the highest variance. The upper bound of this initial value can be roughly featured by the expression O($\sqrt{n}$).

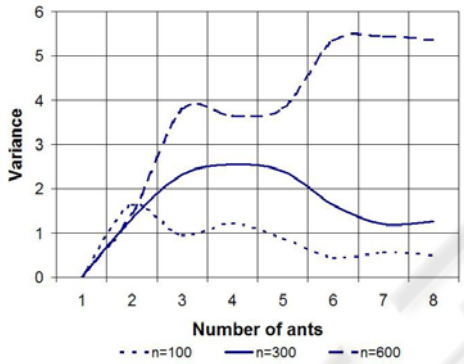

Figure 4: Variance G_1 size 100, 300, and 600.

In the second part of our study we investigated the relationship between the output and the input of the problem, i.e. the most compact achievable embedding for a given set of parameters including the capacity (c) of stores in G_2, as well as, the weight ($\omega(v_i)$) and the order (n) of nodes in G_1. Table 3 shows the parameters under testing and their corresponding levels. Each individual row represents a different experiment.

A simple analysis will indicate that the least number of stores has, as lower bound, the aggregated weight of all documental nodes, divided by the individual store capacity. In other words,

$$stores \geq \frac{\sum_{i=1}^{n} \omega(v_i)}{c} \forall v_i \in V_1$$

In this particular case, we can approximate the summation in the above formula, since we now that $\omega(v_i)$ follows a uniform random distribution between $[a,b]$. Therefore, for any i, $\omega(v_i)$ can be approximated by its mean $(a+b)/2$. Which, in turn, produces

$$stores \geq \frac{n(\frac{a+b}{2})}{c}$$

The sixth column in table 3 shows the lower bound on the number of stores, for each of the experiments consider in this part of our study. Meanwhile, the fifth column shows the mean value obtained with our simulations. Notice that the problem does not have solution when $\omega(v_i)$ can be bigger than c.

In the third group of experiments we addressed the influence of the evaporation factor on the number of rounds. A high evaporation implies a low correlation between the outcomes of consecutive rounds and vice versa. In other words, evaporation features the memory of the previous findings. This time, we tried two evaporation strategies: In the first case, we worked with a fixed factor equal to 0.9. In the second case, we tried with an initial evaporation equal to 0.9 which was decreased by 0.1 on each new round. Table 3 shows the parameters under testing and their corresponding levels. Again, each individual row represents a different experiment. Columns 7 and 8 show the corresponding time complexities for case 1 and 2, respectively. It is quite clear that the second approach is always better than the first one. This new strategy means that we allow a broad initial quest but, as rounds go by, the long term memory prevails and we stick to the best finding to accelerate convergence.

For the second strategy, we evaluated the covariance (S) between n and the number of rounds. The result S =399 indicates that there is a direct dependency between the order of G_1 and the time complexity. In contrast, the covariance between c and the number of rounds is S = -394.41, which means an inverse dependency between the storage capacity and the time complexity.

Table 3: Number of stores and rounds.

Ants (Z)	n	c	$\omega(v_i)$	Number stores	Ideal number of stores	Number rounds FE=0.9	Number rounds variable FE	Number rounds by multiple linear regression
10	100	100	0-20	11.31	10.0	10.37	6.42	7.4598
			0-50	26.15	25.0	12.65	8.03	7.6398
			0-100	53.75	50.0	14.63	9.14	7.9398
		300	0-20	4.46	3.33	8.16	3.81	6.2798
			0-50	9.25	8.33	9.22	4.76	6.4598
			0-100	17.85	16.67	10.64	5.72	6.7598
		900	0-20	2.06	1.11	6.45	2.23	2.7398
			0-50	3.84	2.78	7.32	2.80	2.9198
			0-100	6.35	5.56	8.68	3.38	3.2198
17	300	100	0-60	93.42	90.0	18.18	12.23	8.4998
			0-150	-	-	-	-	-
			0-300	-	-	-	-	-
		300	0-60	32.45	30.0	11.23	6.47	7.3198
			0-150	79.26	75.0	13.24	8.10	7.8598
			0-300	152.89	150.0	16.36	9.74	8.7598
		900	0-60	13.45	10.0	8.42	4.28	3.7798
			0-150	27.69	25.0	9.88	5.36	4.3198
			0-300	53.64	50.0	11.04	6.42	5.2198
30	900	100	0-180	-	-	-	-	-
			0-450	-	-	-	-	-
			0-900	-	-	-	-	-
		300	0-180	275.18	270.0	14.62	9.63	10.4398
			0-450	-	-	-	-	-
			0-900	-	-	-	-	-
		900	0-180	93.64	90.0	10.76	6.87	6.8998
			0-450	228.59	225.0	14.89	8.6	8.5198
			0-900	455.42	450.0	15.98	10.35	11.2198

We assumed there is a linear model that may describe the time complexity as a function of c, n and $\omega(v_i)$, then we used the multiple linear regression model and obtained the following expression

$$rounds = 7.5298 + 0.0040n - 0.0059c + 0.0120(\frac{a+b}{2})$$

The last column in table 3 shows the predicted time complexity for the second strategy, according to this function. We obtained a correlation coefficient equal to 73% between simulation results and the predicted values, which we consider acceptable, for our goals.

7 CONCLUSIONS

We have presented a general methodology that enables the operation of a Corporate Semantic Web (CSW) on top of a P2P distributed storage system. Our semantic-layer proposal is based on two procedures called location and query. Each peer working as a store has a look-up table that supports query. Contents are ordered according to a semantic index. Then, the look-up table shows the peer in charge of the contents associated to a given index. Nevertheless, the main contribution of this work is the location procedure that assigns the contents of the CSW to the corresponding store-peers.

A key hypothesis that may cause debate is that we assume nodes in G_2, i.e. stores, as static entities or, at least, with lifetimes sufficiently long to validate this assumption. In contrast, many empirical studies on networks' dynamics tend to show that unless storage is supported by fixed capacities, cooperative storage is very volatile. Nevertheless, these very studies consider that high information availability can be achieved, even in P2P environments, by means of data replication. Studies (Rodrigues, 2005) suggest that when devices offer a long-term stable service, systems might profit from “conservative” block-coding. In the other side, those systems where devices offer an intermittent service should be built on the bases of “aggressive” replication. Intermediate solutions, with combined replication and block-coding techniques, are also suggested in order to facilitate information retrieval and tracking.

Therefore, we assume a layered architecture and consider our proposal working on a “semantic” layer on top of a “replication” layer. From this perspective, the higher layer works with static logic-stores supported by a lower layer dealing with dynamic, best-effort, storage peers.

Location is modelled in terms of the graph embedding of G_1 into G_2. Here, G_1 represents a CSW and G_2 is the P2P system. Graph embedding (GE) is an NP-complete problem that we tackled

using the Ant Colony Optimization heuristics (ACO). We evaluated the quality and complexities of our location method. As ACO consists of ants or agents that explore the solution space and cyclically improve the results, we found the best number of agents that produce the most efficient exploration. Each round consists of two phases, traversal and prizing. Next, we devised a prizing mechanism that accelerates convergence. For the instances of GE here addressed, we were able to propose a model that predicts the total number of rounds as a linear function of each of the parameters under study.

Some important issues remain as directions for further work. For the sake of simplicity we assumed that each document in the CSW is labelled with a single index. What should we do with multi-labelled or highly nested contents? How should we deal with the CSW growing? Preliminary work suggest that, multi-labelled documents can fit well in our look-up table, by means of linking mechanisms subordinating all the indices of a document to a couple of main concepts that define the actual location of the corresponding file. As for the CSW dynamics, we consider that storage capacities must be kept in order to foresee a middle-term growing. In the long term, it might be the case that the whole CSW partitioning, i.e. its granularity, should be redefined and a new allocation procedure might be invoked. It is also possible that whenever a small subset of related concepts shows a rapid growing on the size of their documents, the entire collection might migrate to a new store node.

Distributed storage is driving many R&D efforts. From the users' point of view, it may turn into the basic mechanism able to unleash the potential benefits of knowledge management. Health sciences, agriculture, geomatics, are only a few examples of the many domains that may dramatically improve their operations with the adoption of this new trend.

REFERENCES

Berners-Lee, T., Hendler, J., Lassila, O., 2001. The semantic web. *Scientific American*.

Cai, M., Frank, M., 2004. RDFPeers: A Scalable Distributed RDF Repository Based on a Structured Peer-to-Peer Network. *In Proceedings of the 13th international conference on the World Wide Web, New York, USA.*

Cidon, I. 1988. Yet Another Distributed Depth-First-Search Algorithm. *Inf. Process. Lett. 26(6)*

Crespo, A. Garcia-Molina, H. 2002. Semantic Overlay Networks for P2P Systems. *Technical report, Stanford University.*

Dorigo, M. 1992. Optimization, Learning and Natural Algorithms. *Ph.D. Thesis, Dept. of Electronics, Politecnico di Milano. Italy.*

Gandon, Fabien. 2002. ONTOLOGY ENGINEERING: A SURVEY AND A RETURN ON EXPERIENCE, *Report of research INRIA, team ACACIA.*

Gutjahr W. 1999. A generalized convergence result for the graph-based ant system metaheuristic. *Technical Report 99-09, University of Vienna.*

Halevy A. Y., Ives Z. G., Mork P., and Tatarinov I. 2003. Piazza: Data management infrastructure for semantic web applications. *In Proceedings of the Twelfth International World Wide Web Conference (WWW'2003), Budapest, Hungary.*

Kjetil, N., Christos, D., Michalis, V. 2006. The SOWES Approach to P2P Web Search Using Semantic Overlays. *WWW '06: Proceedings of the 15th international conference on World Wide Web.*

Montresor, A. 2001. Anthill: a Framework for the Design and the Analysis of Peer-to-Peer Systems. *4th European Research Seminar on Advances in Distributed Systems*.

Nejdl, W. Wolf,B. Qu, C. Decker, S. Sintek, M. Naeve, A. Nilsson, M. Palmer, M. Risch, T. 2002. EDUTELLA: a P2P Networking Infrastructure based on RDF. *In Proceedings of the 11th International World Wide Web Conference, Hawaii, USA.*

Tang, C., Z. Xu, and M. Mahalingam. 2002. PeerSearch: Efficient Information retrieval in Peer-Peer Networks. *Hewlett-Packard Labs: Palo Alto*.

Risson, J, Moors, T. 2006. "Survey of research towards robust peer-to-peer networks: search methods", *Computer Networks: The International Journal of Computer and Telecommunications Networking, Volume 50 , Issue 17, pp 3485-3521.*

Rodrigues R. and Liskov B. 2005. High Availability in DHT's: Erasure Coding vs Replication. IPTPS. 226-239.

Savage J. E. & Wloka M. G.. 1991. MOB a parallel heuristic for graph embedding, *5th SIAM Conference on Parallel Processing for Scientific Computing.*

Segall A. 1983. Distributed network protocols. *IEEE Transaction on Information Theory IT-29(1):23.*

Wolf-Tilo Balke, W Nejdl, W Siberski, and U Thaden. 2005. DL meets P2P - Distributed Document Retrieval based on Classification and Content. *European Conference on Digital Libraries (ECDL), Vienna, Austria*.