DISTRIBUTED SEARCH-BASED ADVERTISING ON THE WEB

Nikita Schmidt and Ahmed Patel[1]

Abstract

A distributed system for Web search and search-based advertising is proposed as a way to solve the scalability problems of centralised Web search and to enable low cost entry to the search and advertising market for service providers. Challenges and potential solutions for implementing search-based document advertising in a distributed search system are discussed. Security and trust issues in a non-cooperative system with distributed ownership and management are outlined. The discussion is based on the experience drawn from implementation and pilot operation of such a system.

1. Introduction

The huge size and continuing growth of the Web have made it difficult for traditional centralised Web search systems to provide complete, relevant, and up-to-date information in response to user search queries. The resources required to download, index, and search all publicly accessible documents on the Web have become for many a prohibitively expensive entry barrier to the Web search market. In addition, the 'invisible' Web, for example non-Web document collections having a Web front-end [1], usually cannot be indexed by centralised search systems.

Distributed search engine architectures [11, 12, 13] have emerged as a solution to these problems. In distributed search architectures, multiple search engines owned by different organisations or individuals act as a single search system.

Search-based advertising, also known as 'pay for placement', is an advertising technique on the Web which uses search systems to target advertisements to the appropriate audience. Content providers who wish to advertise their documents submit their content to search engines for inclusion, usually paying a fee. Search engines return links to advertised documents in response to user search requests that may be relevant to the advertised content.

A system that combines distributed search with advertising has a potential of supporting a number of innovative business and service scenarios. Modest resource requirements for maintaining a single component of the system encourage individuals and organisations to participate, whether as search or advertising service providers, or as advertisers [10].

This paper discusses challenges and solutions arising from implementation and deployment of a distributed search and advertising system. The discussion is illustrated by example of the ADSA system.

[1]Department of Computer Science, University College Dublin, Ireland

2. Background: ADSA system overview

The ADSA (Adaptive Distributed Search and Advertising) project is developing a distributed system of search engines for the Web that support *search-based document advertising*, or *placement*. The main reasons why ADSA is distributed are as follows.

- A distributed design has the potential to address the scalability problem; today, centralised search engines for the Web have to possess enormous resources in order to provide efficient service.
- The system supports *independent ownership and administration* of its individual components, allowing both co-operative and competitive strategies to be employed by individual component owners. Low resource requirements of individual ADSA components create an opportunity for a low cost entry into a global integrated search and advertisement services market.
- Owners of private databases with publicly available search interfaces can make them accessible through the global ADSA network, addressing the problem of 'hidden', or 'invisible' Web.

The rationale behind the project is discussed in detail by Khoussainov et al. [10]. They also present a preliminary architecture which served as a reference during the early stages of the project. In this paper we take a look at how these concepts evolved in the course of the project. We concentrate on the advertising facilities in ADSA, problems encountered and solutions proposed. This section introduces the final system architecture, which is referred to from the rest of the paper.

In the ADSA system there are three basic user types: *searchers*, *advertisers*, and *service providers*. A searcher is any user who wishes to locate documents using the system. An advertiser is any user who wishes to disseminate documents through search-based advertising. A service provider is anyone who wishes to provide a document search and/or advertising service.

Figure 1 shows a high level view of the ADSA system as a distributed collection of *search engines* and *request brokers*. A search engine in ADSA, like any traditional Web search engine, maintains a local index of a set of documents, for which it provides a search service. A search engine may also provide advertisement service, which allows users to place their documents into the search engine's index. The purpose of the broker is to hide the distributed nature of the ADSA system, providing its users with a coherent view of the system as a whole rather than as a loose collection of individual search engines. A broker selects appropriate search engines according to user requests and propagates those requests to the selected engines. To do this, it maintains an index of ADSA search engines.

The arrows in Figure 1 depict document search or advertisement requests. First, a *Service User* (Searcher or Advertiser) sends a request to his/her local Broker. The Broker then determines which Search Engines can best service this request. The request is propagated to the selected Search Engines. The Search Engines process the request by searching for the described documents or by placing the specified documents into their indexes. Each Search Engine sends the results back to the Broker. The Broker collates the results and presents them to the user.

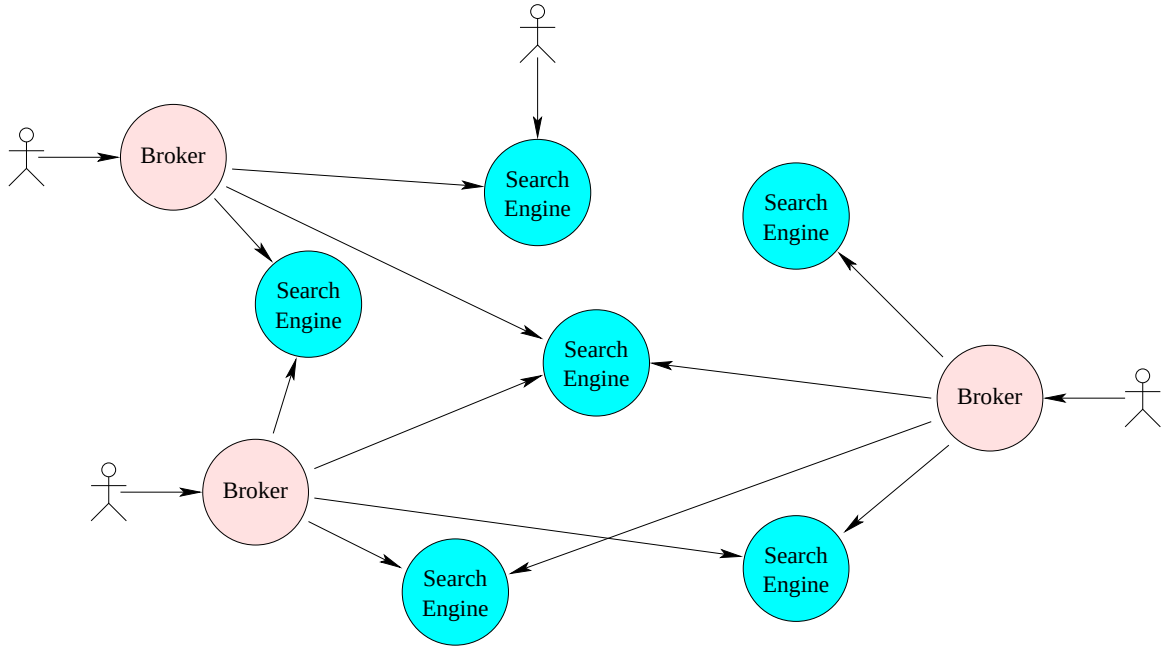

Figure 1: Top level ADSA system view

The picture in Figure 1 is drawn from the viewpoint of the system users: Service Users (Searchers and Advertisers) and Service Providers. Typically, service users send their requests to Brokers, which then automatically propagate those requests to the best Search Engines. Alternatively, service users can also send requests directly to Search Engines (provided that they know their network addresses). Service providers operate Brokers and Search Engines, all of which can be independently owned and maintained.

Search-based advertising is provided by placing relevant advertised documents into the list of search results returned to users. Their rank may be boosted to ensure their prominence on the list. Search-based advertisement will typically be offered for a fee, thus being the primary means of revenue generation in ADSA.

2.1. System components

In the ADSA architecture, brokers and search engines are further split into *components* as shown in Figure 2. Components are units of distribution. Each component can run on a separate computer system, communicating with other components over a TCP/IP-based network. (More than one component can run on the same computer system.) Figure 2 shows top-level request flows in user-component and component-component interactions. Requests in each flow are sequentially numbered.

As illustrated in the diagram, the current architecture provides for five types of components: Document Database (DDB), Service Directory (SDIR), Search Client (SCL), Advertisement Client (ACL), and Administration (or Management) Client (MCL). Search and Advertisement Clients in the broker provide distributed search to their users by first consulting the Service Directory and then making requests to search engines (more precisely, to their Document Database components). The same clients in the search engine provide simpler functionality by going

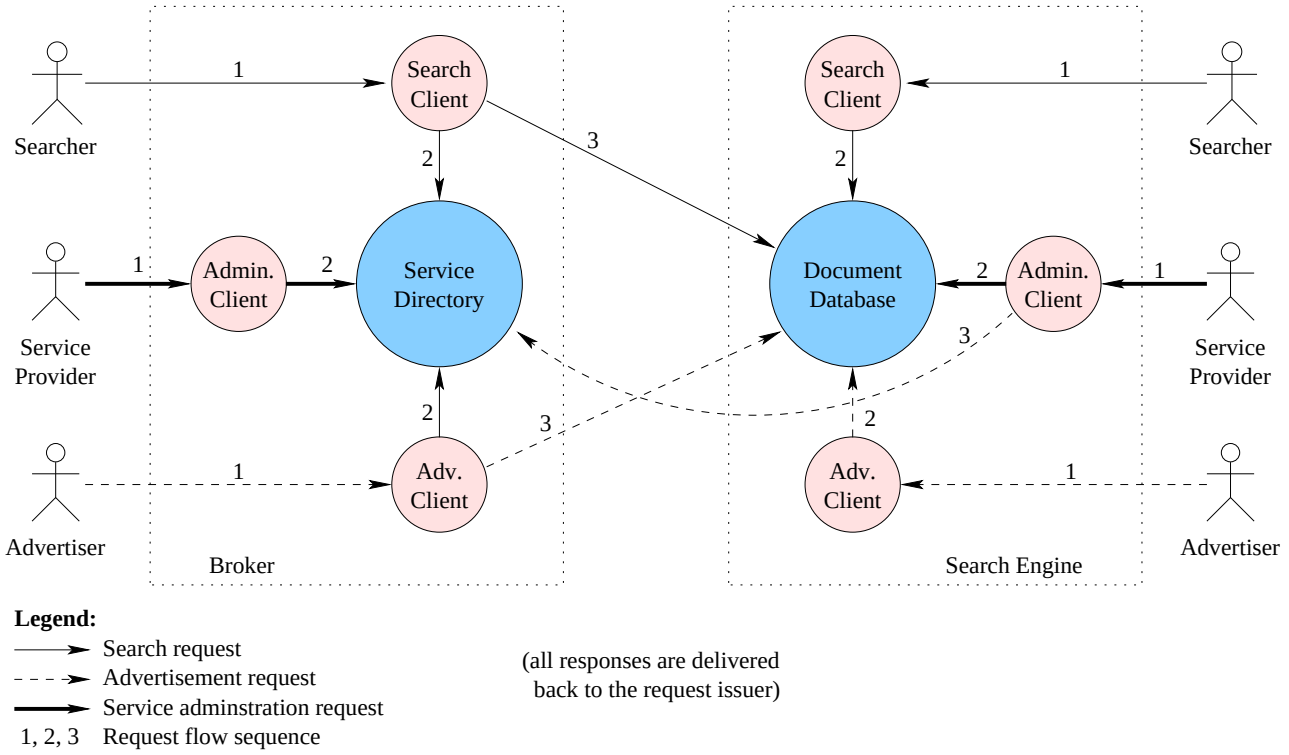

Figure 2: System architecture overview and component breakdown

directly to the Document Database of their search engine. This allows search service providers for local (on-site) content to give their users direct access to their search engine (for example, by linking search client's interface page off their main Web pages). This is how private search engines operate today.

Note that the system is distributed at the component level rather than the broker/search engine level. This provides much greater flexibility on at least two counts:

- Each service provider's installation is flexible and can itself be distributed. For example, a provider running search engines can use several computers for the databases, and use one Administration Client to manage them all together. Or, a service provider may not want to run search and/or administration clients at all.
- Each individual component can be independently owned and administrated, which opens up the potential for innovative service and business scenarios.

All ADSA components are briefly described in the following subsections.

2.1.1. Document Database

A Document Database is the essence of an ADSA search engine, which provides a 'search engine'-like interface to its indices for search and/or advertising. A Document Database can use its own topic-specific Web robot to populate itself with documents from the Web. It can also

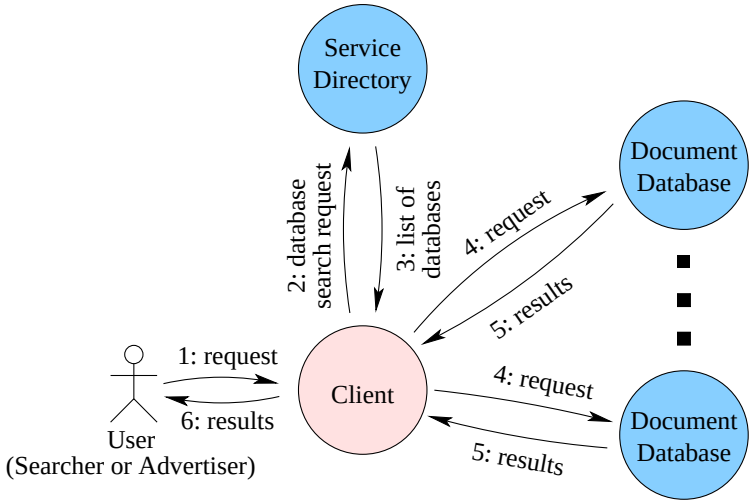

Figure 3: Overview of search and advertising scenarios

serve its local content, loaded by the Database administrator. A Document Database responds to keyword-based queries by providing ranked lists of document URLs from its storage. It also responds to document advertising (placement) requests by indexing the submitted documents, thereby providing a search service for the advertised documents.

2.1.2. Service Directory

A Service Directory can be called a 'database of databases': its purpose is to find best known Document Databases that provide the required services for a given search or advertisement request. Databases can advertise (list) themselves in Service Directories to become globally known to the ADSA system. Like the Document Database, it has a 'search engine style' interface: a Directory accepts requests in the same (or almost the same) form as a Document Database would, but returns a ranked list of Database names rather than document URLs.

There are well-known database selection algorithms based on topical similarity, such as GlOSS (Glossary-of-Servers Server) [8, 9], CORI (Collection Retrieval Inference Network) [4], CVV [14]. Database selection methods in a Web environment are investigated in [6].

2.1.3. Search and Advertisement Clients

The Search and Advertisement Clients provide the system with a Web-based user interface (HTML over HTTP). The clients communicate with Directories and Databases in order to satisfy user requests according to the user requirements. Clients would typically submit a user's request to a selected Directory, retrieve a list of best matching services (Document Databases), and re-submit the request to one or more of the Databases, as illustrated in Figure 3. The results returned from the Database(s) will be passed onto the user. A Client, therefore, performs request brokerage and query propagation using information from a Directory. The Clients also maintain user accounts.

2.1.4. Administration Client

The Administration Client is responsible for service management and administration of the Document Database and Service Directory system components. Its functions are adjusting parameters of running Databases and Directories, and advertising Databases in Directories.

3. Issues with distributed advertising

This section discusses various issues and research problems related to search-based advertising in a distributed system, using ADSA as an example.

3.1. Business strategy: topic and price selection

Efficient operation of a distributed search system can only be achieved if most individual document databases are topic-specific, i.e., each one specialises in its own topic. This is necessary for efficient query routing.

The choice of a topic and a pricing policy is a business decision which depends on the market conditions, for example:

- topics catered for by other document databases, both in and outside of ADSA;
- quality of competing search and advertisement services;
- costs of maintaining a certain service quality on a given topic;
- prices of the competition.

A possible approach to creating an automated or semi-automated decision support system for advertising in ADSA is to employ game-theoretic techniques [11].

3.2. Boosting of advertised documents

In order to meet its service obligations or simply to increase its revenue, a document database may want to increase the prominence of advertised documents in the result lists returned to the user. The following aspects of such prominence boosting must be considered:

- the effect of boosting on the likelihood of the user choosing the advertised document;
- the decrease of search quality;
- the service obligations of the database with respect to its advertisers.

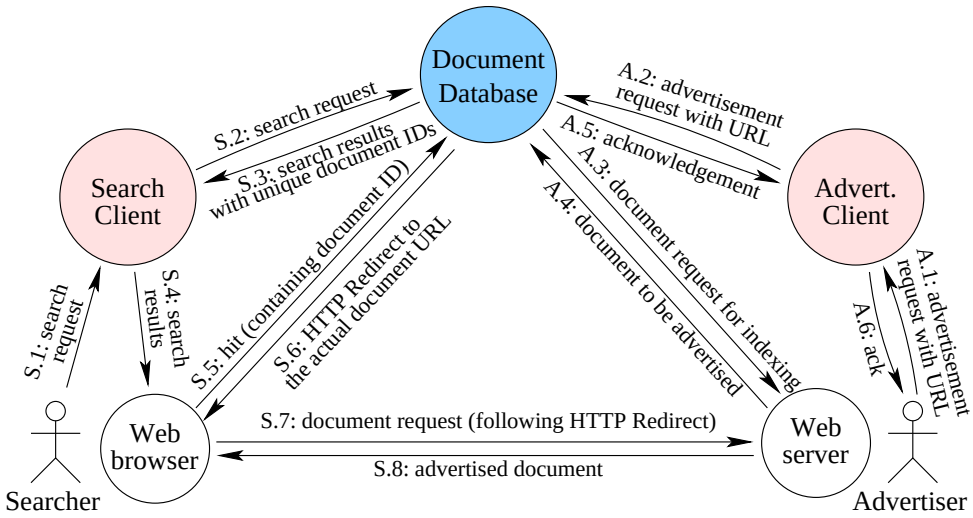

Figure 4: Search and advertising scenario illustrating hit counting

The trade-offs such as search quality versus advertising performance need to be carefully considered.

The need for boosting depends on the topic of the advertised document: the closer the document topic is to the topic of the database, the less boosting it may need. Indirect costs of boosting (such as decline in revenue due to search service deterioration) may need to be taken into account. Thus, a database may assign prices to each advertised document individually depending on its topic in order to reflect indirect costs associated with its advertising.

3.3. Hit counting

Hit counts provide a good measurable characteristic of the business performance of a document advertising service. Hit count is the number of 'hits' an advertisement has generated, i.e., the number of times users followed the advertised link returned with search results. Hit counts are important for the database: they provide accounting information and also allow the database to evaluate and adjust its boosting strategy.

Hits can be counted using the standard redirect technique. Instead of returning links to advertised documents in its search results, a database returns hyperlinks pointing to itself and carrying a unique document ID as a parameter. When the user follows such a link, the database records the hit, looks up the ID and returns an HTTP Redirect [7] containing the actual document URL.

A scenario when a document is advertised and then requested by a search user is illustrated in Figure 4. Transactions between the Clients and the Service Directory are omitted for brevity. Requests and responses for document advertising are marked A.1–A.6, and those for search are marked S.1–S.8.

A document database needs to be able to correspond a hit to the search request the hit came from. This information is important for analysing the performance of boosting. Thus, each

search request must generate new unique reference IDs for all advertised documents that made their way into search results.

3.4. Security and trust issues

The questions of trust and protection of information are especially important in competitive environments. A document database has to co-operate with other components such as directories and clients; at the same time, it wants to protect its information, such as topic, pricing, quality of service provided, and business performance, from the competition. These goals are sometimes contradictory.

Topic and price have to be registered with a service directory in order for the database to receive any search queries and advertising business at all. This information is public and is readily available to the competition. An attempt of a database to lie about its topic to the directory can be detected by the directory via probe requests, for example, using *query-based sampling* [2, 3].

Quality of service may be sampled by firing probe requests at the database in question. This is more troublesome than getting information from a directory, but still quite doable and there is no effective way to guard against such probing.

An advertiser may lie to the database he or she advertises in by supplying a fake document for indexing and then replacing the link target with another document. Such behaviour may have an impact on the database's search quality. Thus, a database may want to occasionally verify advertised documents by re-downloading them.

A malicious entity (e.g., a competing document database) may want to fool a database into recording fictitious hits. This can be done by generating a stream of requests that look like those generated by the user's Web browser when the user follows an advertised link. (Such requests normally result in the database recording a hit and returning a redirect to the actual document.) To protect itself against such attacks, a database should ensure that document reference IDs are not generated in a predictable way.

4. Discussion

We have identified three parameters that determine the performance of a document advertising service in a distributed non-cooperative system. These parameters are topic, pricing policy, and boosting strategy. The choice of these parameters depends on competition, search queries received from users, and feedback in the form of hit counts.

In order for hit counting to be effective, the reference IDs assigned to advertised documents for the purposes of hit tracking must be unique for each search request and each advertised document. Since hit counts are used for both feedback and accounting purposes, at least the following information should be recorded for each hit: time; reference to the corresponding search request; reference to the advertised document; and the source of the hit (e.g., the IP address of the user's Web browser).

Attention to trust and protection issues is an important survival skill in a competitive environment. From the technical point of view, it means that the following measures must be taken or at least considered:

- a service directory must probe the databases registered with it to confirm that their service descriptions (topics, prices, etc.) are correct;
- a document database must probe the documents advertised with it to confirm that their content has not changed significantly;
- a document database must be prepared that information such as its topic, pricing policy, and hit counts may be publicly available, and that its service quality and boosting strategies may be probed by competition;
- random document reference IDs generated for the purposes of hit counting are better than predictable (e.g., sequential), as this helps prevent spoof attacks.

These conclusions have been drawn from the experience of design, implementation, and operation of the ADSA system.

5. Conclusion

A distributed system for Web search and search-based advertising is proposed as a way to solve the scalability problems of centralised Web search and to enable low cost entry to the search and advertising market for service providers. The system allows for both co-operative and non-cooperative strategies and creates a platform for innovative business and service scenarios.

This paper discusses technical issues associated with implementing and deploying search-based document advertising services in such a system. The two most important technical issues are hit counting and trust. Three business model issues — topic and price selection and boosting strategy definition — are identified but not discussed in detail.

As a model of a distributed system for this discussion we chose the ADSA system, briefly described in the paper. This system was in pilot service for more than a year, which has served as a test bed and a validating platform for the technical issues discussed in the paper.

Two further research directions can be identified:

- solving the business model challenges, such as automatic topic, price and boost management;
- developing an integrated security model in collaboration with current research on security in autonomic computing and communication environments [5].

6. Acknowledgements

The support of the Informatics Research Initiative of Enterprise Ireland for funding the ADSA project is gratefully acknowledged. We also wish to acknowledge Science Foundation Ireland for funding under the NTSRC development grant, particularly for the security and trust aspects of this work.

References

[1] BAILEY, P., CRASWELL, N., and HAWKING, D., Dark matter on the Web, in: Proceedings of the 9th International World Wide Web Conference, Amsterdam, The Netherlands 2000.

[2] CALLAN, J., CONNELL, M., and DU, A., Automatic discovery of language models for text databases, in: Proceedings of the ACM SIGMOD 1999 International Conference on Management of Data, vol. 28/2, pp. 479–490, ACM Press 1999.

[3] CALLAN, J., POWELL, A.L., FRENCH, J.C., and CONNELL, M., The effects of query-based sampling on automatic database selection algorithms, Technical Report CMU-LTI-00-162, Language Technologies Institute, School of Computer Science, Carnegie Mellon University 2000.

[4] CALLAN, J., LU, Z., and CROFT, W.B., Searching distributed collections with inference networks, in: Proceedings of the 18th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval, pp. 21–28, ACM Press 1995.

[5] CHESS, D.M., PALMER, C.C., and WHITE, S.R., Security in an autonomic computing environment, in: IBM Systems Journal, 42(1):107–118 (2003).

[6] CRASWELL, N., BAILEY, P., and HAWKING, D., Server selection on the World Wide Web, in: Proceedings of the 5th ACM Conference on Digital Libraries, pp. 37–46, San Antonio, Texas, USA 2000.

[7] FIELDING, R.T., GETTYS, J., MOGUL, J.C., NIELSEN, H.F., MASINTER, L., LEACH, P.J., and BERNERS-LEE, T., Hypertext transfer protocol — HTTP/1.1, RFC 2616, Internet Engineering Task Force 1999. `http://www.ietf.org/rfc/rfc2616.txt`.

[8] GRAVANO, L. and GARCÍA-MOLINA, H., Generalizing GlOSS to vector-space databases and broker hierarchies, in: Proceedings of the 21st International Conference on Very Large Data Bases (VLDB '95), pp. 78–89 (1995).

[9] GRAVANO, L. and GARCÍA-MOLINA, H., GlOSS: Text-source discovery over the Internet, in: ACM Transactions on Database Systems, 24(2):229–264 (1999).

[10] KHOUSSAINOV, R., O'MEARA, T., and PATEL, A., Adaptive distributed search and advertising for WWW, in: N. Callaos et al. (ed.), Proceedings of the 5th World Multi-Conference on Systemics, Cybernetics and Informatics (SCI 2001), vol. 5, pp. 73–78, Orlando, Florida, USA 2001.

[11] KHOUSSAINOV, R., O'MEARA, T., and PATEL, A., Independent proprietorship and competition in distributed Web search architectures, in: Proceedings of the 7th IEEE International Conference on Engineering of Complex Computer Systems (ICECCS 2001), pp. 191–199, Skövde, Sweden 2001.

[12] SUEL, T., MATHUR, C., WU, J., ZHANG, J., DELIS, A., KHARRAZI, M., LONG, X., and SHANMUGASUNDERAM, K., ODISSEA: A peer-to-peer architecture for scalable Web search and information retrieval, WWW2003 poster (2003).

[13] WATERHOUSE, S., JXTA search: Distributed search for distributed networks, Technical report, Sun Microsystems, Inc., Palo Alto, CA, USA 2001.

[14] YUWONO, B. and LEE, D.L., Search and ranking algorithms for locating resources on the World Wide Web, in: Proceedings of the 12th International Conference on Data Engineering, pp. 164–171, IEEE Computer Society 1996.