Service Description in a Distributed Search and Advertising System

NIKITA SCHMIDT and AHMED PATEL
Department of Computer Science
University College Dublin
Belfield, Dublin 4
IRELAND

Abstract: Service description in a distributed system allows system components to know about one another and to make intelligent decisions regarding request routing and propagation. The paper discusses service description model used in a distributed system for Web search and search-based advertising. A service description consists of content description (terms), attribute set, and a number of service parameters (price, size, speed, etc.). The model addresses such features of the system as integration with invisible Web, attribute-based search and advertising, and pay for placement. The impact of the model on scalability is discussed.

Key-Words: Distributed search architecture, Search-based advertising, Hidden Web, Attribute, Security

1 Introduction

Distributed search architectures [6, 11] have the potential to address two issues becoming increasingly prominent in today's World Wide Web:

- spreading the load of indexing and searching vast amounts of informations (such as the Web) across different organisations and individuals;
- allowing independent search service providers access to the global Web search market.

As the Web size grows, it is becoming increasingly difficult for a single individual or organisation to provide a quality search service for the entire Web. This requires extensive equipment and bandwidth investments. Distributing the indexing and search load across many organisations significantly decreases such requirements for each individual organisation.

Allowing independent search service providers to take part in a distributed search system fosters development of innovative search services. The cost of entry to a distributed search market is much lower than that of using a centralised architecture.

The "invisible Web" problem [10] is also addressed. The invisible, or hidden, Web consists of information providers who cannot make all their information available to a third-party search engine for indexing. In a distributed scenario, they can implement their own search interface to their data and plug it into the distributed search network.

A typical problem of distributed search architectures is that of *query routing*. Its aim is to ensure that each user query reaches the relevant search service provider. Query brokers, responsible for routing, need to know descriptions of each provider, which allow them to decide which providers are suitable for each query. This paper discusses service descriptions used in the ADSA project, which has successfully completed its pilot service operation.

2 ADSA Architecture and Features

The ADSA project[1] set out to build a working prototype of a distributed search system for the WWW. The goal was to enable independent ownership of the system components, which together provide an integrated search service. This was to address the issues mentioned above, namely scalability, cost of access to the Web search services market, and invisible Web.

The system has a two-tier architecture. A user query comes to a *query broker*, which finds the best *search service*, forwards the query to the service (which can be thought of as a small-scale centralised search engine), and returns the results back to the user.

[1] `http://cnds.ucd.ie/adsa/`

The system is designed as a set of *components*. A component is a unit of distribution, capable of running independently on a network-connected machine. There are five types of component: *Document Database* (an individual search engine), *Service Directory* (responsible for finding Document Databases as part of the query broker functionality), *Search Client* (providing user interface and query forwarding), *Advertisement Client* and *Management Client*. The latter two are not discussed further in this paper. Document Databases provide *search services*, while Search Clients together with Service Directories provide *query brokerage services* (Fig. 1).

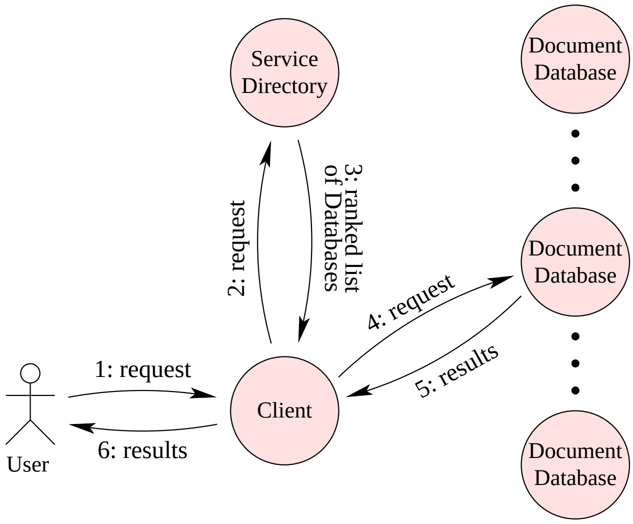

Fig. 1: Search scenario

2.1 Additional features

Topic-specific Web crawler. For a distributed system to be effective, a Service Directory must be able to differentiate Document Databases on the basis of a user query. Since the query typically consists of search keywords, it means that distribution of keywords among Databases must be uneven—i.e., Databases should specialise in different topics. This is achieved by a topic-specific (focused) Web crawler [7].

Document placement (advertising). In order to make the system attractive to prospective participants, it provides a facility to implement *search-based advertising* services. Such a service allows users to place (advertise) their documents in Document Databases. Links to these documents are then returned amongst search results in response to relevant search queries. Providers (ADSA participants) may charge their users for this service. The placement scenario is very similar to search scenario shown in Fig. 1, except that the client in question is an Advertisement Client, the request contains the document to advertise (rather than search terms), and the response indicates success or failure.

Attribute-based search. This feature permits the use of attributes (such as ‘author’, ‘title’, ‘ISBN’, ‘model’, ‘brand’, etc.) in search queries together with search terms. Many custom search interfaces, which constitute the invisible Web, provide such a facility to increase search precision. This makes it even more difficult for them to integrate into a global search system because of the inherent differences among attribute sets in different domains. For instance, attributes used for book search (e.g. ‘title’ and ‘ISBN’) will likely have nothing in common with those for car search (e.g. ‘fuel type’ and ‘transmission’). To support integration of invisible Web, ADSA architecture allows each Document Database to have its own, flexibly configurable attribute set [9]. This requires special considerations with respect to query routing, as will be discussed below.

3 Service Registration

To become part of the system, a Document Database must make itself known to a Service Directory. This is called *service advertisement* (registration). A Database provides a document search service and may also provide a document advertisement service. The description of each service is submitted to one or more Service Directories (Fig. 2). Because service descriptions change, they must be updated (re-submitted) on a regular basis.

Because ADSA allows competition between independent component owners, it is possible that a Document Database may decide to submit false service descriptions in order to unfairly boost its prominence. To protect itself from such an attack, a Service Directory may employ query sampling [2, 5].

4 Service Description

The goal of a service description is to allow a Service Directory to decide which Database is best for a given search query or document advertisement re-

Element	**Content**	**Description**
Content description	List of (term, frequency) pairs	This is a representation of the *topic* of a Database. Each word (term) indexed by the Database is given together with the number of times it occurs in all indexed documents. Terms are stemmed and exclude stop words.
Attribute set	List of attribute names	Search attributes are identified by their names. This list includes *content attributes* (those pertaining to the content of a document, e.g., 'title'), *metadata attributes* (describing properties of a document, e.g., 'location', or URL), and *special attributes* (those which affect the processing of a search or advertising request, e.g., 'example'—the URL of an example document).
Service parameters	List of (name, value) pairs	Price, number of documents indexed, search speed, etc.

Table 1: Service description

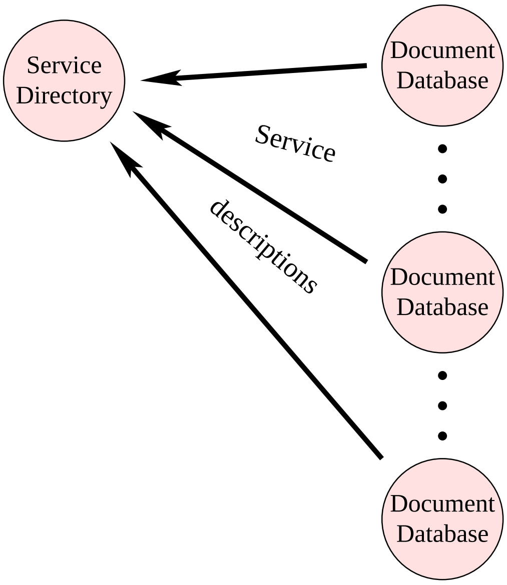

Fig. 2: Service registration

quest. Table 1 describes items that constitute service description in the ADSA architecture.

Content description allows Service Directory to calculate Database relevance based on the search keywords in the user query (for a search request), or on the contents of the document being advertised (for a placement request). This is suitable for a variety of well-known collection selection algorithms such as GlOSS [3, 4], CORI [1] and CVV [12].

Note that the form of the content description as a list of terms and their frequencies is intentionally "raw". It could be possible to use a more "processed" and compact representation—for example, through some topic classification system such as the Dewey decimal system, or through LSI (latent semantic indexing) "features". However, the intention was to allow different competing implementations of the Service Directory. Thus, interfaces between components must be kept independent of implementation.

5 Service Ranking

The list of services returned from a Service Directory to the Search Client is ranked according to their relevance to the user request. This relevance is multi-dimensional, as it takes the following into account:

- *Relevance of request content to Database content.* This is computed using standard collection selection algorithms.

- *Similarity of request attribute set to Database attribute set.* Most queries do not use attributes, in which case this dimension is not considered. However, if a query is attribute-based, this similarity becomes very important. Firstly, attribute set is characteristic of the domain (e.g., books or cars) where search or advertisement is being conducted. Secondly, a service that supports attributes specified in a user query stands a better chance of servicing that request well.

- *Constraints on service price and other parameters.* Users may specify such constraints along with their requests. Even when all constraints are satisfied, a cheaper service should receive a higher ranking than a more expensive one of similar relevance.

6 Security and Privacy

Issues of security and privacy cannot be ignored in a system that supports paid services in a cooperative/competitive environment. Among such issues are:

- *Trust:* components in the system are generally untrusted, but may form trusted coalitions.
- *Confidentiality:* financial data and trade secrets (such as pricing strategies and advertising algorithms) need protection.
- *Authentication:* a reliable method to identify trusted components and entities with which certain agreements have been made.

These must be carefully balanced against the requirement to maintain a certain level of cooperation, without which the system would lose cohesion and fall apart. For example, in its service description a service cannot hide or lie about its price—otherwise it will not be found or will have to reject user requests based on wrong assumptions implied by the service description. For a similar reason, it makes no sense to hide the attribute set. Misrepresentation of content description can be dealt with by the Service Directory through query sampling as described in section 3.

Service descriptions are affected by security and privacy issues in two aspects. The first aspect is the protection of information contained in the description. While most of the information is public, certain confidential service parameters may be added to it when it is submitted to a suitable trusted component. For that, the service (Document Database) must authenticate the remote component, verify that it is trusted, and use an encrypted connection to protect confidentiality in transit.

The second aspect is allowing the user to search for services that have specific security properties (e.g., the ability to establish a secure connection). For that, such properties must be advertised in a service description. For instance, a military Database may indicate its classification, which will also imply the level of security protection that will be in place when accessing the Database.

The current ADSA prototype does not support security features in service descriptions. However, it supports encrypted communication and authentication. Adding more flexibility at this stage will require too much human administrative effort to manage it; it will simply be ignored. Reducing this effort is a subject for further research in collaboration with security-related studies in the context of autonomic communicating systems.

7 Scaling

With ADSA being a two-tier architecture, a question arises as to how scalable it is now and how difficult it would be to lift the two-tier limitation. Simulation and pilot trials have shown that for both Databases and Directories, performance is mostly affected by the number of documents/services indexed, rather than the size of a document or service description. The Document Database implemented in the prototype shows acceptable (sub-second in most cases) performance on collections of up to 100 000 documents on high-end commodity hardware [8]. The Service Directory has never been tested with more than 857 service descriptions. If it scales as well, then ADSA can potentially support one Directory indexing 100 000 Databases, each indexing 100 000 documents, for a total of ten thousand million documents. This is about twice the number of documents indexed by Google today.

These figures suggest that, in reality, a two-tier architecture is stretching its limits with the current size of the Web. So why is the two-tier limitation present in the ADSA model and how can it be removed?

Adding depth to the architecture requires the ability to advertise a Service Directory in another Service Directory. For that, it must be able to generate its service description. Let us take a look at what that description may look like.

Content description will naturally be the sum of all content descriptions submitted to the Directory. While this is not an obstacle in itself, it will only be useful as long as Directories themselves are topic-specific to some extent.

Attribute set will need to evolve from a simple list of attribute names to a list of (attribute, frequency) pairs. Frequencies will be used in attribute relevance calculations. Attribute frequencies in Database service descriptions will all be set to 1. Again, this will only be useful as long as services with similar attribute sets are advertised in the same Directory.

The other service parameters (price, number of documents, speed, etc.) will need to be replaced with their statistical data such as minimum, maximum, median, mean and standard deviation. These will provide a reasonable estimation of the kind of services indexed by a Directory. The less the deviation, the more useful these data are.

Finally, a Directory will have its own service parameters—related to the service *it provides*, rather than the services *indexed*. These will include price for search in Directory, number of services indexed, etc.

These questions have not been approached by the current prototype and are subject for further investigation.

8 Conclusion

The service description model described here is based on the results of a successfully completed research and development project, which built a distributed systems for search and advertising. The model addresses various system usage scenarios, including distributed search, pay for placement, invisible Web, and attribute-based search. To support these scenarios, a service description consists of three parts: content description, attribute set, and service parameters, which are discussed in detail.

The model is independent of any particular implementation of service indexing. This facilitates flexibility in developing different algorithms while retaining compatibility among system components.

One problem associated with the presented service description model is that of scalability beyond the two-tier 'broker–search engine' architecture, potentially up to unlimited broker nesting. A direction towards a possible solution is suggested in the paper. However, it has not been tested in the ADSA project and its system prototype, and is left for further research.

Another topic for further investigation is security aspects of service descriptions. Related research is being conducted on security in autonomic communications environments.

9 Acknowledgements

The authors would like to acknowledge the support of Enterprise Ireland for funding the ADSA project, and Science Foundation Ireland for partly sponsoring this research.

References:

[1] J. P. Callan, Z. Lu, and W. B. Croft, 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, July 1995.

[2] J. Callan, A. L. Powell, J. C. French, and M. Connell, 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.

[3] L. Gravano and H. García-Molina, 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, Sept. 1995.

[4] L. Gravano and H. García-Molina, GlOSS: Text-source discovery over the Internet, *ACM Transactions on Database Systems*, Vol. 24, June 1999, pp. 229–264.

[5] P. G. Ipeirotis and L. Gravano, Distributed search over the hidden-web: Hierarchical database sampling and selection, in *Proceedings of the 28th International Conference on Very Large Data Bases (VLDB 2002)*, 2002.

[6] R. Khoussainov, T. O'Meara, and A. Patel, 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)*, (University of Skövde, Skövde, Sweden), pp. 191–199, IEEE Computer Society Press, Los Alamitos, California, USA, 11–13 June 2001.

[7] T. O'Meara and A. Patel, A topic-specific Web robot model based on restless bandits, *IEEE Internet Computing*, Vol. 5, Mar./Apr. 2001, pp. 27–35.

[8] T. Phelan, A. Patel, and S. Ó. Ciardhuáin, Simulation based approach to evaluate a distributed search engine, in *Proceedings of the IADIS International WWW/Internet 2003 Conference*

(P. Isaías and N. Karmakar, eds.), Vol. I, (Algarve, Portugal), pp. 347–354, International Association for Development of the Information Society, IADIS Press, 5–8 Nov. 2003.

[9] N. Schmidt and A. Patel, Distributed search for structured documents, in *Proceedings of the 8th Australian World Wide Web Conference (AusWeb 02)* (A. Treloar and A. Ellis, eds.), (Sunshine Coast, Australia), pp. 256–273, Southern Cross University, July 2002. `http://ausweb.scu.edu.au/aw02/papers/refereed/patel/`.

[10] C. Sherman and G. Price, *The Invisible Web: Uncovering Information Sources Search Engines Can't See*. Medford: CyberAge Books, Dec. 2001.

[11] S. Waterhouse, JXTA search: Distributed search for distributed networks, tech. rep., Sun Microsystems, Inc., Palo Alto, CA, USA, May 2001.

[12] B. Yuwono and D. L. Lee, 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, Feb. 1996.