Opportunities from Open Source Search

Wray Buntine
Complex Systems Computation Group,
Helsinki Institute for Information Technology
P.O. Box 9800, FIN-02015 HUT, Finland
wray.buntine@hiit.fi

Karl Aberer, Ivana Podnar and Martin Rajman
School of Computer and Communication Sciences
EPFL, Lausanne, Switzerland

Abstract

Internet search has a strong business model that permits a free service to users, so it is difficult to see why, if at all, there should be open source offerings as well. This paper first discusses open source search, and a rationale for the computer science community at large to get involved. Because there is no shortage of core open source components for at least some of the tasks involved, the Alvis Consortium is building infrastructure for open source search engines using peer-to-peer and subject specific technology as its core, based on this rationale. We view open source search as a rich future playground in which information extraction and retrieval components can be used and intelligent agents can operate.

1 Introduction

Since search has a business model that permits a free service to users it is difficult to see why, if at all, there should be open source offerings as well. In this paper we first discuss the background of search, so that we may present a rationale that makes open source search a realistic development for the open source and research community. We then describe the infrastructure that the Alvis Consortium is developing based on this rationale. Alvis is a semantic based, peer-to-peer (P2P) search system under development.

1.1 Search and Information

Search is driven by advertising and is a business with billions of dollars in annual revenue in 2004 using pay-for-placement and targeted keyword advertising models. Google, the major player, is now the world's largest media company by stock market value and the number three business-to-business company, along side established companies like the Wall Street Journal. Newspaper and broadcast television has had a slow decline and the Internet is now viewed as a creditable information source with a substantial and increasing viewer-ship[1] in many areas, including news and current affairs.

Search is essential for navigation in the Internet, just like directories such as Yahoo were in the early stages, and dominant players have emerged due to consolidation in the top end. The business of keyword search is a natural monopoly just like operating systems: "size matters" when it comes to coverage and response time. Analysts believe that no new monolithic search engine for the Internet as a whole can emerge because the scale of investment and development required is too great.

1.2 Open Source Offerings

Major search engines do a reasonable job, they are free to users, and they are constantly innovating. Why should open source search be made available? Before considering this question, we review what is currently available. The best publicised system in recent years is Nutch[2], now part of the Apache Foundation. This is the basis for several subject specific search sites such as the Creative Commons search engine[3], and is part of the Lucene project. Another recent system is Terrier which has stronger foundations in current research [16]. Open source search goes back at least to ht://Dig[4], released in 1995, and many other systems targeting intranet use including MySQL-based systems have been released before and since. In the related text processing areas of information extraction and natural language pro-

[1] http://www.stateofthemedia.org/2005
[2] http://www.nutch.org
[3] http://creativecommons.org/find/
[4] http://www.htdig.org

cessing, systems often remain proprietary but in recent years increasing numbers of systems are seeing release in open source or use (e.g., using Library GPL).

Traditional open source arguments applied to Internet search go as follows: Internet search is now approaching a monoculture, where largely secret recipes are used to return results. This is suboptimal from a technology perspective, we need different ranking algorithms and methods for different needs. Providing an alternative is beneficial *per se*. The near monoculture is also against the whole spirit of open source and its counterpart in the media world called open media [15] typified by bloggers. Note that in desktop search, because of the Linux platform, no such near monoculture exists.

1.3 A Challenge

Why else should open source search be pursued? Some computer scientists argue that *intelligent searching of the Internet* is a task that holds pride of place with high profile computing challenges such as chess, soccer playing robots, and space exploration. It is a problem of international scope and clear need that has its origins and its solutions firmly in computer and information science.

To let researchers access this grand challenge we need an open source search engine operating at a larger scale. Such a platform would not only serve as an excellent research and educational tool, it could also support a wide variety of applications and act as an important commodity to cost-conscious organisations that provide services.

1.4 A Niche for Open Source

As in the Microsoft vs. Linux battle over operating systems, we have strong arguments for wanting open source alternatives in some specific areas. Opportunities exist where a combination of new technology and subject focus can combine to create a better service. One parallel business does this already. The so-called Enterprise Content Management community touts phrases such as “better return from your digital assets” to corporate executives.

The open source world usually develops in a low cost environment. Some communities that use specialised search services are as follows:

Alternative Languages: some languages challenge keyword search due to their rich morphology (e.g., Estonian, Slovenian, Turkish) or their lack of clear word segmentation (e.g., Chinese).

Digital Libraries: libraries require richer user interfaces and better document and access control than the standard search engine.

Publishing Initiatives: open publishing, open archive, open media and open access initiatives on the Internet foster varied distribution of content.

Academic Special Interest Groups: academics have their own document genres and sometimes rich ontologies.

Blogs: several blog search engines already exist, but opportunities for social network studies, trend and topic detection detection, etc., are there.

It is in and for these communities that robust development of search engines exists outside the mainstream. Analysis of these communities reveals the potential for incorporating additional capability into a search engine such as subject categories, genre, named entities, and question answering tools. In digital library applications, for instance, this kind of feedback and capability is valued [7].

2 Rationale for Open Source Search

Open source search will only provide a useful service if it has capabilities not easily found at the global search engines. The capabilities of experts within their own domains should be better employed to customise subject specific search engines. For instance, the “Environmental Search Engine” could provide services unique to its own domain such as carefully developed lists of corporations, pollutants and species, relevant subject categories, better selection of material, and cataloguing of material under genres, etc.

Open source search can target small-scale alternatives where individual commercial incentives are inadequate. Open source search could then leverage its products across many such alternatives using P2P techniques. With the right architecture and standards, the grand challenge of intelligent search then becomes accessible to computer scientists in many institutions. Of course, there is also the potential here for a new environment for competitive business development, not unlike the Linux world or the early Internet.

Alvis[5] is a European research project building infrastructure for semantic-based P2P search in an open source environment. A consortium of eleven partners from six different European Community countries plus Switzerland and China contribute expertise in a broad range of specialities including network topologies, routing algorithms, probabilistic approaches to information retrieval, linguistic analysis and bioinformatics. Two of the most promising research areas applicable to search the Alvis group see are P2P systems [3, 14] and information extraction (IE).

Information extraction is described by the GATE group[6] as processing “unrestricted text in order to extract information about pre-specified types of events, entities or relationships”. Recently, IE has been shown to be able to develop

[5] http://www.alvis.info

[6] http://gate.ac.uk/ie/

simple ISA hierarchies [13], used for instance by Semantic Web systems. Thus information extraction can semi-automate the task of tagging Internet content, the task that is generally considered a major hurdle for semantic web progress [4]. IE is also the basis for recent state of the art question answering systems that turn text content into a knowledge base to be subsequently explored using fast indexing [5].

P2P systems, subject specific search engines, and information extraction are three complementary technologies. Information extraction tools provide the technology to help domain experts customise smaller search engines without large investments of time or sophisticated programming efforts; P2P systems become more efficient at information retrieval when the nodes are topically oriented [10]; and, P2P systems provide the technology to make some sense out of a network of smaller search engines, to let users have a single access point to a network of such services.

3 Infrastructure

In the Alvis project we have begun the design of an infrastructure to support this kind of effort. Our overriding design principle is that we need to have an open architecture and standard interfaces. We want diverse experts to contribute components to the system and software to be integrated such as information retrieval systems, Terrier, Lucene, and Lemur, information extraction systems such as MALLET [12] and GATE, and enumerable tools for shallow parsing and crawling.

There are many ways of viewing search engines, but a general decomposition we have arrived at which takes into account the additional needs of information extraction and P2P querying is given in Figure 1.

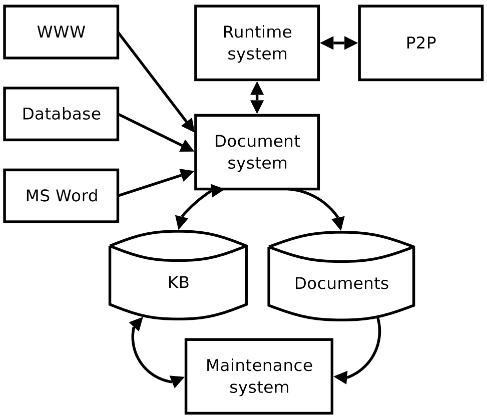

Figure 1. Overall architecture

Document system: starting with the crawl, general document processing and creation of content for the runtime system, for instance using information extraction tools.

Maintenance system: collection level processing to develop linguistic and semantic resources such as gazetteers, category systems, and genres, for instance with ontology discovery. This maintains the knowledge base (KB).

Runtime system: an indexing system and query-retrieval system that would be viewed as a single node in the P2P network, which we call a superpeer. Includes *query processing*, *generating snippets* and other *per-document data for display*, and the *end user interface*.

Peer-to-peer: While the superpeers are stand-alone search engines for their own content, the P2P services offer network wide query and retrieval. The runtime system enters document and content information and supports P2P services as requested.

For these general functions, we need standard formats that allow independent components to be added and integrated as needed without tight procedural or functional integration.

3.1 Document System

Where possible, we borrowed well developed standards that could serve some role in the system. Data and metadata for a single document is carried in XML with elements for the different processed components (original document in HEX, crawler details, semantic tagging, etc.). Companion XSL transformations support different internal tasks such as link processing and data extraction for different stages of the document pipeline. Entries in the XML format include where necessary:

RDF: our simple ontologies will be represented using RDF[7], when ontology terms need to be incorporated in metadata, for instance, topical entries from the Open Directory Project.

Canonical Document Format: general structural processing of content cannot be written to operate on all kinds of material, PDF, email, Word, etc., thus we developed a simplified structural format containing just sections, lists and links. Documents are first converted into this format and the general processing applies to this. The original document is kept for reference purposes.

Linguistic Annotation: natural language processing (NLP) of any kind results in annotations made to a document, for instance to tag a noun phrase as a *person*, or to give the basic dictionary forms or

[7] http://www.w3.org/RDF/

lemma of a word and part of speech (e.g., for “ran” or “runs” it would be “run/VERB”). Annotation is a long studied engineering problem in NLP. An ISO proposition (TC37SC4/TEI) is being developed by the NLP community, and we have adopted a simple variation of this [2]. Note linguistic annotation can be very bulky, thus it is kept internally between relevant document and maintenance systems and not exported to the runtime system where only the results such as semantic tags are needed.

The document processing system is then just a pipeline or network of processing steps. While we initially have a simple two-stage pipeline of crawler and basic information extraction, we envisage more complex pipelines in the future, for instance different linguistic or named entity modules and page or site authority ranking modules will be incorporated. The Open Archives Initiative Protocol for Metadata Harvesting[8] (OAI-PMH) is a flexible standard for delivering batches of marked-up documents, and several good open source libraries implement it. It can be used in a straight linear pipeline, or in a more complex processing network with offshoots or side-tracks. It thus supports a distributed approach to processing batches of documents, and is already being used by some larger websites to deliver content to crawlers.

3.2 Runtime System

The runtime system can act as a stand-alone information retrieval engine. Within Alvis, we use the XML-capable information retrieval system Zebra[9], which provides us with more flexible metadata facilities than traditional IR systems but not a full semantic web capability. In principle, however, most of the high quality open source information retrieval engines should serve the same task. The runtime system is also the primary connection point to the P2P system. When documents enter the local runtime system, it is required that they be registered with the P2P system. As needed during querying, the local runtime system also provides document information services to the P2P system.

The common query language[10] (CQL) provides a common format for queries, and several good open source libraries implement it. But we also needed an XML standard for representing search results, so developed one. This format has the ability to incorporate the grouping of documents, auxiliary annotations, and auxiliary categorisations found in many current avant garde search engines. For instance, keyword information, topic categorisation, geographic information, relevant names of people or organisations, or other auxiliary information could be included in this format. XSL can again be used as a definition language for converting our previous document format to this results format.

4 P2P Large-scale Information Retrieval

The Alvis approach to open-source search relies on an underlying structured P2P overlay network [1] for maintaining a distributed inverted index of large-scale document collections. Arguments promoting structured P2P solutions as potential counterparts to centralised systems arise from their inherent properties, namely, *scalability*, *self-organisation*, and *fault-tolerance*. Structured P2P networks limit the routing latency by $O(logN)$ number of hops, while the routing information maintained by each peer is also limited to $O(logN)$, where N is the total number of peers in a network. A recent analysis shows that such logarithmic-style networks exhibit properties of small-world networks capable of supporting non-uniformly distributed resource keys (which is the case in information retrieval), while preserving good load-balancing properties [8]. Open-source large-scale search can particularly benefit from P2P self-organisation and fault-tolerance since P2P systems require minimal in place infrastructure and maintenance, which significantly reduces costs compared to centralised solutions. Moreover, such networks have no centralised authority that can affect system performance, for example, influence the overall ranking process, or choose the collection set for indexing.

The Alvis architecture comprises two types of nodes as depicted in Figure 2: *peers* building the distributed P2P overlay network, and *superpeers*, stand-alone components hosting document collections. The network of peers is responsible for maintaining a distributed index of superpeers’ document collections, and enables efficient querying of its distributed index. Superpeers are capable of performing advanced information retrieval services such as sophisticated processing of document collections to build semantically rich indexes enhanced by various ranking strategies, and to support complex structured queries. They incorporate the Alvis document and runtime system, but are not peers in the sense of P2P.

Superpeers interact with peers using a communication protocol to do the following: to *submit index* of its local document collection for incorporating it into a distributed index, and to *send query* to a distributed P2P index. A peer can therefore be regarded as an entry point to a distributed index, and a P2P overlay network as a scalable and efficient media for sharing data among the superpeers. Different types of superpeers can be incorporated into our architecture, e.g. sophisticated search engines, digital libraries, Internet servers hosting collections of unstructured documents, or even only query-enabled components such as Web browsers.

Note that the P2P overlay network does not store document collections because it is not designed to be a distributed archival storage system such as Oceanstore [17], but a querying system that knows the location of documents related to a query. Furthermore, superpeers have a freedom to design a document set they want to provide to the distributed

[8] http://www.openarchives.org/OAI/openarchivesprotocol.html

[9] http://www.indexdata.dk/zebra/

[10] http://www.loc.gov/z3950/agency/zing/cql/

index. Our querying procedure is designed to be flexible: An answer to a query can be created solely using the information available in the distributed index offering good response times while sacrificing precision. The other approach with good precision and low responsiveness is performed as a two-step process. In the first step, a list of peers containing documents relevant to a query is extracted from the distributed index, and in the second step a query is sent directly to peers, their answers are merged, re-ranked, and submitted to the originating querying component. Using the second approach users can benefit from special complex querying procedures supported by superpeers, and superpeers can decide whether they want to provide such services to the open public.

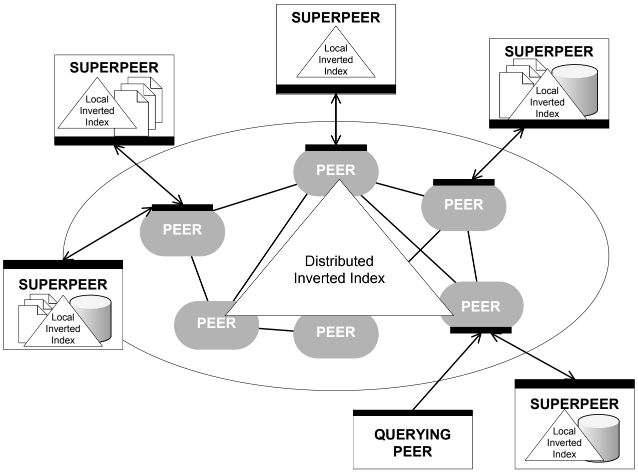

Figure 2. Architecture for P2P information retrieval

P2P overlay builds a distributed index of $(key, data)$ pairs offering key-based routing: $insert(key, data)$ routes the data to a peer responsible for a given key, and $search(key)$ retrieves the data for a given key. On the contrary, superpeers are dealing with $(term, postinglist)$ pairs, where a posting list identifies documents containing a term together with document-related term statistics. A naïve approach to distributing a superpeer's index, and transforming it to a $(key, data)$ pair, hashes a term to produce a key, and encodes a posting list into a data field. The approach may work for relatively small document collections, but cannot be applied to large-scale document collections. Analysis presented in [9] shows that the naïve approach is infeasible due to unacceptable storage and traffic requirements caused by extremely large posting lists for the Internet-scale document collection. It is therefore necessary to design special algorithms and techniques to implement, and deploy a workable solution comparable in performance to centralised systems.

Although the area P2P information retrieval is still in its infancy, a number of potential optimisation techniques have been identified, such as caching, compression, document clustering, and semantic space reduction. Our efforts for optimisation can broadly be classified into two categories, algorithms for transforming and mapping superpeer's local index to P2P distributed index, and methods to improve P2P network performance for the particular area of information retrieval.

Algorithms for mapping inverted index to key-based P2P index. Our solution for index mapping is quite intuitive as depicted in Figure 3. We are limiting the size of posting lists by expanding the vocabulary which now contains context-related 'rare keys' (sets of terms). Such rare keys are highly discriminative since they occur in a limited number of documents, therefore we call them Highly Discriminative Keys (HDKs). Experimental evaluation shows that, when carefully choosing rare keys, the growth of the HDK vocabulary and HDK index remains linear with respect to collection size. Subject-specific document collections and special coding techniques can further reduce the size of the HDK index. To address the problem of finding relevant keys for a given query, we use an approach based on *distributional semantics* where we calculate a co-occurrence matrix to make a probabilistic connection between the full vocabulary and our HDK vocabulary. Detailed explanation of the HDK approach with complexity analysis and experimental evaluation can be found in [11].

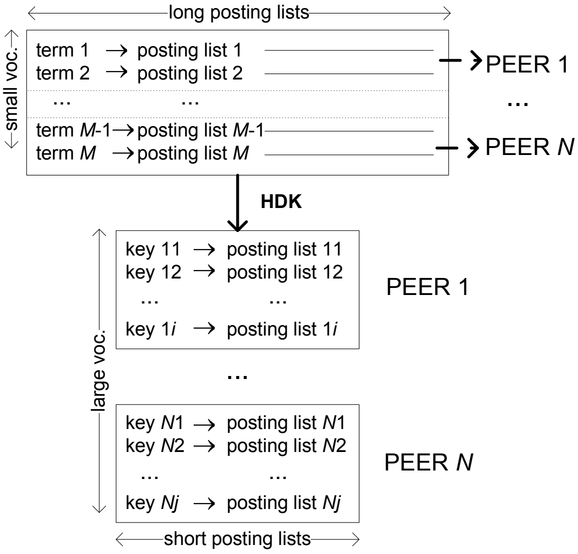

Figure 3. The basic idea of indexing using HDKs

P2P optimisation methods. Since the overall system is highly-dependent on the performance of the underlying P2P implementation, we are working on various optimisation methods to tune up P-Grid [19], our P2P implementation, for specific requirements imposed by information retrieval. We are dealing with a large amount of data that needs to be indexed, however, indexing is not time-critical and should generate minimal amount of network traffic. On the contrary, system responsiveness is critical for querying.

Initial results show the potential of our approach to preserve a retrieval quality (top-k precision) comparable to the standard single term TF*IDF approach. This allows scaling to a level centralised systems cannot reach, while algorithmic complexity and bandwidth consumption remain acceptable. An almost unlimited storage space becomes available provided that enough peers are integrated in the network, a process which require minimal infrastructure and maintenance.

5 Opportunities

Opportunities abound once the right infrastructure is in place. Here we described just some of the kinds of systems that could exist in such a framework, intended either as academic or commercial projects. These opportunities would be the greatest benefit of open source search to our community.

5.1 A Trust/Reputation Consortium

On Google the ranking of pages is influenced by the PageRank^{TM} of websites. Sites appearing in the top 10 results for certain queries get a significant boost in viewership, and thus PageRank^{TM} becomes critical for marketing purposes. This method for computing authority for a web page borrows from early citation analysis, and the broader fields of trust, reputation, and social networks (which blog links could be interpreted to represent) provide new opportunities for this kind of input to search. Analysis of large and complex networks such as the Internet is readily done on todays grid computing networks.

What are some scenarios for the use of new kinds of data about authority, trust and reputation, standards set up by a consortium perhaps. A related example is the new OpenID[11], a distributed identity system.

ACM could develop a ”computer science site rank” that gives web sites an authority ranking according to ”computer science” relevance and reputation. In this ranking the BBC Sports website would be low, Donald Knuth's home page high, and Amazon's Computer Science pages somewhat high. Our search engines can then incorporate this authority ranking into their own scores when asked to do so. ACM might pay for the development and maintenance of this ranking as a service to its members, possibly incorporating its rich information about citations as well. In an open source search network, consumers of these kinds of organisational or professional ranks could be found.

Yahoo could develop a vendor web site classification that records all websites according to whether they primarily or secondarily perform retail or wholesale services, product information, or product service. This could be coupled with a vendor login service so that venders can manage their entries, and trust capabilities so that some measure of authority exists about the classifications. Using this, search engines then have a trustworthy way of placing web pages into different product genres, and thus commercial and product search could be far more predictable.

5.2 The Genre Directory

Search user studies divide users according to their purpose or goal [18]. The logical next step is to consider the function of web-pages. If users have different purposes or goals for their search, these should be matched by the function of the documents returned. Function of a document can be characterised by its genre. According to the Merriam-Webster Online Dictionary, *genre* is a category of artistic, musical, or literary composition characterised by a particular style, form, or content. One can define genre in terms of the purpose of a document, but looking for a way to automatically detect the purpose of a document leads back to style. Web genre is quiet different to print media, and typical categories include [6] FAQs, link collections, dedicated multiparty correspondence, private and informal content, public or commercial, etc.

Genre is intrinsically difficult to detect in web pages, thus only major distinctions are supported to date, typically general web, news, blogs, and scientific literature. Developing directories of genre and providing categorisation tools for genre would allow far finer support for user's functional needs. Our search engines could then use this at the user interface, or for corpus filtering.

Acknowledgements

This work is supported by the EU 6th Framework Programme in the IST Priority under the ALVIS project. The material borrows many of the ideas from the Alvis Consortium's submissions to the EU's 6th Framework Programme, and benefits from interactions with the Alvis partners. Thanks also to Ville Tuulos and Aleks Jakulin for feedback.

References

[1] K. Aberer, M. Hauswirth, M. Punceva, and R. Schmidt. Improving Data Access in P2P Systems. *IEEE Internet Computing*, 6(1), 2002.

[11] http://www.openid.net/

[2] E. Alphonse, S. Aubin, J. Derivière, T. Hamon, D. Mladenic, A. Nazarenko, C. Nédellec, T. Poibeau, D. Weissenbacher, and Q. Zhou. Report on method and language for the production of augmented document representations. Alvis Deliverable D5.1, Alvis, 2004.

[3] J. Callan and N. Fuhr. The SIGIR peer-to-peer information retrieval workshop. *SIGIR Forum*, 38(2), 2004.

[4] S. Cherry. Weaving a web of ideas. *IEEE Spectrum*, 39(9):65–69, 2002.

[5] T. Clifton and W. Teahan. Knowing-aboutness: Question-answering using a logic-based framework. In *ECIR 2005*, pages 230–244, 2005.

[6] J. Dewe, J. Karlgren, and I. Bretan. Assembling a balanced corpus from the internet. In *11th Nordic Computational Linguistics Conference*, 1998.

[7] O. Drori. How to display search results in digital libraries-user study. In *New Developments in Digital Libraries, 3rd Int. Workshop*, pages 13–28, 2003.

[8] S. Girdzijauskas, A. Datta, and K. Aberer. On Small World Graphs in Non-uniformly Distributed Key Spaces. In *Proceedings of the 1st IEEE International Workshop on Networking Meets Databases (NetDB)*, 2005.

[9] J. Li, T. Loo, J. Hellerstein, F. Kaashoek, D. Karger, and R. Morris. On the Feasibility of Peer-to-Peer Web Indexing and Search. *IPTPS03*, 2003.

[10] J. Lu and J. Callan. Content-based retrieval in hybrid peer-to-peer networks. In *CIKM*, pages 199–206, 2003.

[11] T. Luu, F. Klemm, M. Rajman, and K. Aberer. Using Highly Discriminative Keys for Indexing in a Peer-to-Peer Full-Text Retrieval System. Technical Report 2005041, School of Computer and Communication Sciences, Ecole Polytechnique Fédérale de Lausanne (EPFL), 2005.

[12] A. McCallum. MALLET: A machine learning for language toolkit. http://mallet.cs.umass.edu, 2002.

[13] C. Nédellec. Ontologies and information extraction. In S. Staab and R. Studer, editors, *Handbook on Ontologies in Information Systems*. Springer Verlag, 2004.

[14] W. Nejdl. How to build Google2Google - an (incomplete) recipe. In *International Semantic Web Conference*, pages 1–5, 2004. Invited talk.

[15] T. Perkins. Introducing the AO/Technorati open media 100. *AlwaysOn*, Issue 06-23-05, 2005.

[16] V. Plachouras, B. He, and I. Ounis. Experiments in web, robust and terabyte tracks with Terrier. In *Proceedings of the Thirteenth Text REtrieval Conference (TREC 2004)*, 2005.

[17] S. Rhea, C. Wells, P. Eaton, D. Geels, B. Zhao, H. Weatherspoon, and J. Kubiatowicz. Maintenance-free Global Data Storage. *IEEE Internet Computing*, 5(5), September/October 2001.

[18] D. Rose and D. Levinson. Understanding user goals in web search. In *WWW '04: Proceedings of the 13th international conference on World Wide Web*, pages 13–19, 2004.

[19] The P-Grid Consortium. The P-Grid project, 2005. http://www.p-grid.org/.