How to Build Google2Google
– An (Incomplete) Recipe –

Wolfgang Nejdl

L3S and University of Hannover, Germany
nejdl@l3s.de

Abstract. This talk explores aspects relevant for peer-to-peer search infrastructures, which we think are better suited to semantic web search than centralized approaches. It does so in the form of an (incomplete) cookbook recipe, listing necessary ingredients for putting together a distributed search infrastructure. The reader has to be aware, though, that many of these ingredients are research questions rather than solutions, and that it needs quite a few more research papers on these aspects before we can really cook and serve the final infrastructure. We'll include appropriate references as examples for the aspects discussed (with some bias to our own work at L3S), though a complete literature overview would go well beyond cookbook recipe length limits.

1 Introduction

Why should we even think about building more powerful peer-to-peer search infrastructures? Isn't Google sufficient for every purpose we can imagine? Well, there are some areas where a centralized search engine cannot or might not want to go, for example the hidden web or community-driven search.

The hidden web requires replication of data usually stored in databases in a central search engine which seems like a bad idea, even though Froogle (http://froogle.google.com/) attempts to do this for a limited domain / purpose (shopping, of course). A central data warehouse just does not seem an appropriate infrastructure for the world wide web, even though replicating much of the surface web on the Google cluster is (still) doable.

The main characteristic of a community-driven search infrastructure is its strong search bias on specific topics / results. While the last two years have seen techniques emerging for biasing and personalizing search [1,2,3], catering for a lot of small and specific search communities in a centralized search engine neither seems easy to accomplish nor particularily useful to implement for a search engine company which has to target the average user, not specialized communities.

Besides covering these new areas, another advantage of a distributed search infrastructure is the potential for faster updates of indices, because we can exploit local knowledge about new and updated content directly at the site which provides it, without necessarily crawling all of its content again. Last, but not least, decentralized search services might also appeal to those who would rather opt for a more "democratic" and decentralized search service infrastructure instead of centralized services provided by a (beneficial) monopolist or a few oligopolists.

S.A. McIlraith et al. (Eds.): ISWC 2004, LNCS 3298, pp. 1–5, 2004.

In this talk, we will discuss some necessary ingredients for a Google2Google recipe, which have to be mixed together to provide a general decentralized search service infrastructure. Please be aware, that a well-tasting solution is still a few years away, and cooking it all together is not as simple as it might seem at the first moment.

2 Distributed Search Engine Peers

As the first ingredient, we need a large set of distributed Google2Google search peers. These are not just distributed crawlers such as the peers in the distributed search engine project Grub (http://grub.org/), but rather provide crawling, indexing, ranking and (peer-to-peer) query answering and forwarding functionalities.

Each Google2Google search peer will be responsible for a small partition of the web graph, with some overlaps to achieve redundancy, but without centralized schedulers. As the web graph is block structured, with inter-block links much sparser than links within blocks, search peers have to orient themselves on this structure [4,5].

Obviously, it will be important how we connect these peers. A super peer architecture [6,7] might be a good choice because it allows us to include task specific indexing and route optimization capabilities in these super peers. As for the exact topology, we will have a range of options, most probably building upon one of the P2P topologies derived from Cayley graphs [8] used in DHT and other P2P networks [9,10,11,12].

Typical search engine users will rely on simple text string searches, though for more sophisticated applications we certainly want to provide more sophisticated query capabilities as they are available for example in the Edutella network [13]. For certain applications we might opt for publish/subscribe infrastructures instead [14], which can re-use many of the ingredients mentioned in this talk.

3 Distributed Ranking Algorithms

One of the main reasons Google quickly replaced older search engines was its ability to rank results based on the implicit recommendations of Web page writers expressed in the link structure of the web. So we definitively have to throw in a ranking algorithm to provide Google2Google with comparable functionalities.

Unfortunately, two main ingredients of Google's ranking algorithm (TF/IDF and PageRank) rely on collection wide document properties (IDF) and central computation (PageRank). There is hope, however, that we are able to solve these problems in the future: IDF values in many cases do not change much when new documents are added [15], and distributed algorithms for computing pagerank and personalized pagerank have been proposed [16,3].

Furthermore, pagerank variants more suited to decentralized computation have recently been investigated [17,18], and promise to decrease communication costs in a Google2Google setting. These algorithms compute PageRank-like values in two separate steps, first doing local computation within a site, then computation between sites (without taking specific pages into account). Additional analysis is needed on how they compare to PageRank and how sites with a lot of non-related pages (e.g. mass hosters like Geocities) can be handled. Google2Google search peers will probably have to rely on blocks within these sites or group the pages of such sites in community-oriented clusters.

4 Top-k Retrieval and Optimization

Obviously, ranking algorithms are not enough, we also have to use them to prune Google2Google answers to the best k ones. Imagine shipping hundred thousand answers and more to the user, who only wants to look at the top ten or top twenty answers in most cases anyway.

Recent work on top-k query processing and optimization in peer-to-peer networks has addressed these issues, and has shown how (meta) data on query statistics and ranking methods can be used to retrieve only the k best resources [19,20]. Ranking in this context is used to score the results that come from distributed sources, reduce the number of answers, enable partial matches to avoid empty result sets and optimize query forwarding and answering. Briefly, these algorithms assume a super-peer network connected using for example a hypercube-derived topology [12], and implement three intertwined functionalities (more details are presented in another presentation [19]):

- Ranking: Each peer locally ranks its resources with respect to the query and returns the local top-k results to its super-peer.
- Merging: At the super-peers results from the assigned peers are ranked again and merged into one top-k list. These answers are returned through the super-peer backbone to the querying peers, with merges at all super-peers involved.
- Routing: Super-peer indices store information from which directions the top-k answers were sent for each query. When a known query arrives at a super-peer these indices are used to forward the query to the most promising (super-) peers only. A small percentage of queries - depending on the volatility of the network - has to be forwarded to other peers as well, to update the indices in a query-driven way.

5 Trust and Security

Finally, trust and security play a role even more important in our decentralized Google2Google infrastructure than in more centralized settings. Topics here range from the issue of decentralized trust establishment and policy-based access control for distributed resources [21,22] to distributed computations of trust values meant to achieve the same benefits as trust networks in social settings [16,23]. The presentations at the 1st Workshop on Trust, Security and Reputation on the Semantic Web [24] covered a large set of topics in this area and initiated a lot of fruitful discussions.

In our Google2Google setting, malicious peers could for example try to subvert information they provide, an issue especially critical for ranking information. Recent analysis of attack scenarios in a distributed PageRank setting [16,3], have been encouraging, though, and have shown that with suitable additional modifications of the distributed algorithm, PageRank computation becomes quite un-susceptible to malicious peers even in a decentralized setting. Personalized PageRank variants can be made even more resiliant if we bias them towards trusted sites / peers [25,26].

6 Conclusion

Even though a decentralized infrastructure makes seemingly simple things surprisingly difficult (for example how to implement the “Did you mean xx” functionality in Google, which relies on (global) query and answer statistics), our short search for available ingredients for a Google2Google infrastructure already turned out rather successful. It will still take quite a few more research papers (plus an appropriate business / incentive model for a distributed search engine infrastructure) until we will be able to finalize our Google2Google recipe. But once we are finished we will have most certainly realized quite a few new opportunities for searching and accessing data on the (semantic) web.

Acknowledgements. Regarding the title of the paper as well as the initial idea, let me thank my friend and colleague Karl Aberer for inventing and using the term Google2Google as a nice catch phrase for distributed web search engines, on occasion of the panel he organized at the 2nd International Workshop on Databases, Information Systems, and Peer-to-Peer Computing [27]. This workshop took place in Toronto on August 29 and 30, in the context of VLDB’04. Regarding the results and ideas described in this paper, I gratefully acknowledge all my collaborators at L3S and elsewhere for their valuable contributions on the issues discussed in this talk and mentioned in the reference list in the next section. Without their help, my cookbook recipe would not have been written.

References

1. Haveliwala, T.: Topic-sensitive pagerank. In: Proceedings of the 11th International World Wide Web Conference, Honolulu, Hawaii. (2002)
2. Jeh, G., Widom, J.: Scaling personalized web search. In: Proceedings of the 12th International World Wide Web Conference, Honolulu, Hawaii, USA (2003)
3. Chirita, P., Nejdl, W., Scurtu, O.: Knowing where to search: Personalized search strategies for peers in P2P networks. In: SIGIR Workshop on Peer-to-Peer Information Retrieval, Sheffield, UK (2004)
4. Cho, J., Garcia-Molina, H.: Parallel crawlers. In: Proceedings of the Semantic Web Workshop, 11th International World Wide Web Conference, Honolulu, Hawaii, USA (2002)
5. Kamvar, S.D., Haveliwala, T.H., Manning, C.D., Golub, G.H.: Exploiting the block structure of the web for computing pagerank. In: Proceedings of the 12th Intl. World Wide Web Conference, Budapest, Hungary (2003)
6. Yang, B., Garcia-Molina, H.: Designing a super-peer network. In: Proceedings of the 19th Internatonal Conference on Data Engineering, Bangalore, India (2003)
7. Nejdl, W., Wolpers, M., Siberski, W., Schmitz, C., Schlosser, M., Brunkhorst, I., Loser, A.: Super-peer-based routing and clustering strategies for RDF-based peer-to-peer networks. In: Proceedings of the 12th International World Wide Web Conference, Budapest, Hungary (2003)
8. Nejdl, W., Qu, C., Kriesell, M.: Cayley DHTs: A group-theoretic framework for analysing DHTs based on cayley graphs. Technical report, University of Hannover (2004) submitted for publication.
9. Ratnasamy, S., Francis, P., Handley, M., Karp, R., Shenker, S.: A scalable content addressable network. In: Proceedings of the 2001 Conference on applications, technologies, architectures, and protocols for computer communications, ACM Press New York, NY, USA (2001)

10. Stoica, I., Morris, R., Karger, D., Kaashoek, M.F., Balakrishnan, H.: Chord: A scalable peer-to-peer lookup service for internet applications. In: Proceedings of the 2001 Conference on applications, technologies, architectures, and protocols for computer communications, ACM Press New York, NY, USA (2001)
11. Aberer, K., Cudré-Mauroux, P., Hauswirth, M.: A framework for semantic gossiping. SIGMOD Record **31** (2002) http://www.p-grid.org/Papers/SIGMOD2002.pdf.
12. Schlosser, M., Sintek, M., Decker, S., Nejdl, W.: HyperCuP—Hypercubes, Ontologies and Efficient Search on P2P Networks. In: International Workshop on Agents and Peer-to-Peer Computing, Bologna, Italy (2002)
13. Nejdl, W., Wolf, B., Qu, C., Decker, S., Sintek, M., Naeve, A., Nilsson, M., Palmér, M., Risch, T.: EDUTELLA: a P2P Networking Infrastructure based on RDF. In: Proceedings of the 11th International World Wide Web Conference, Hawaii, USA (2002) http://edutella.jxta.org/reports/edutella-whitepaper.pdf.
14. Chirita, P.A., Idreos, S., Koubarakis, M., Nejdl, W.: Publish/subscribe for RDF-based P2P networks. In: Proceedings of the 1st European Semantic Web Symposium, Heraklion, Crete (2004)
15. Viles, C., French, J.: On the update of term weights in dynamic information retrieval systems. In: Proceedings of the Fourth International Conference on Information and Knowledge Management, Baltimore, Maryland, USA (1995)
16. Kamvar, S., Schlosser, M., Garcia-Molina, H.: The eigentrust algorithm for reputation management in P2P networks. In: Proceedings of the 12th International World Wide Web Conference. (2003)
17. Wang, DeWitt: Computing pagerank in a distributed internet search system. In: Proceedings of the 30th International Conference on Very Large Databases, Toronto (2004)
18. Wu, J., Aberer, K.: Using siterank for decentralized computation of web document ranking. In: Proceedings of the 3rd Intl. Conference on Adaptive Hypermedia and Adaptive Web-Based Systems, Eindhoven, Netherlands (2004)
19. Nejdl, W., Siberski, W., Thaden, U., Balke, W.T.: Top-k query evaluation for schema-based peer-to-peer networks. In: Proceedings of the 3rd International Semantic Web Conference, Hiroshima, Japan (2004)
20. Balke, W.T., Nejdl, W., Siberski, W., Thaden, U.: Progressive distributed top-k retrieval in peer-to-peer networks. submitted for publication (2004)
21. Yu, T., Winslett, M., Seamons, K.: Supporting Structured Credentials and Sensitive Policies through Interoperable Strategies in Automated Trust Negotiation. ACM Transactions on Information and System Security **6** (2003)
22. Gavriloaie, R., Nejdl, W., Olmedilla, D., Seamons, K., Winslett, M.: No registration needed: How to use declarative policies and negotiation to access sensitive resources on the semantic web. In: Proceedings of the 1st First European Semantic Web Symposium, Heraklion, Greece (2004)
23. Ziegler, C., Lausen, G.: Spreading activation models for trust propagation. In: Proceedings of the IEEE International Conference on e-Technology, e-Commerce, and e-Service. (2004)
24. Bonatti, P., Golbeck, J., Nejdl, W., Winslett, M.: ISWC'04 workshop on trust, security, and reputation on the semantic web. http://trust.mindswap.org/trustWorkshop (2004) Hiroshima, Japan.
25. Gyöngyi, Z., Garcia-Molina, H., Pedersen, J.: Combating web spam with trustrank. In: Proceedings of the 30th International Conference on Very Large Databases, Toronto (2004)
26. Chirita, P., Nejdl, W., Schlosser, M., Scurtu, O.: Personalized reputation management in P2P networks. Technical report, University of Hannover (2004)
27. Ooi, B.C., Ouksel, A., Sartori, C.: The 2nd intl. workshop on databases, information systems and peer-to-peer computing. http://horizonless.ddns.comp.nus.edu.sg/dbisp2p04/ (2004) co-located with VLDB'04, Toronto, Canada.